Eduardo Bayro-Corrochano
Jan-Olof Eklundh (Eds.)

LNCS 5856

Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications

14th Iberoamerican Conference on Pattern Recognition, CIARP 2009
Guadalajara, Jalisco, Mexico, November 2009
Proceedings

Springer

Lecture Notes in Computer Science 5856

Eduardo Bayro-Corrochano
Jan-Olof Eklundh (Eds.)

Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications

14th Iberoamerican Conference
on Pattern Recognition, CIARP 2009
Guadalajara, Jalisco, Mexico, November 15-18, 2009
Proceedings

Springer

Volume Editors

Eduardo Bayro-Corrochano
CINVESTAV, Unidad Guadalajara
Department of Electrical Engineering and Computer Science
Jalisco, México
E-mail: edb@gdl.cinvestav.mx

Jan-Olof Eklundh
KTH - Royal Institute of Technology
Centre for Autonomous Systems
School of Computer Science and Communication
Stockholm, Sweden
E-mail: joe@nada.kth.se

Library of Congress Control Number: 2009937938

CR Subject Classification (1998): I.5, I.4, I.2.10, I.2.7, F.2.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN	0302-9743
ISBN-10	3-642-10267-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-10267-7 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12787630 06/3180 5 4 3 2 1 0

Preface

The 14th Iberoamerican Congress on Pattern Recognition (CIARP 2009, Congreso IberoAmericano de Reconocimiento de Patrones) formed the latest of a now long series of successful meetings arranged by the rapidly growing Iberoamerican pattern recognition community.

The conference was held in Guadalajara, Jalisco, Mexico and organized by the Mexican Association for Computer Vision, Neural Computing and Robotics (MACVNR). It was sponsodred by MACVNR and five other Iberoamerican PR societies. CIARP 2009 was like the previous conferences in the series supported by the International Association for Pattern Recognition (IAPR).

CIARP 2009 attracted participants from all over the world presenting state-of-the-art research on mathematical methods and computing techniques for pattern recognition, computer vision, image and signal analysis, robot vision, and speech recognition, as well as on a wide range of their applications.

This time the conference attracted participants from 23 countries, 9 in Iberoamerica, and 14 from other parts of the world. The total number of submitted papers was 187, and after a serious review process 108 papers were accepted, all of them with a scientific quality above overall mean rating. Sixty-four were selected as oral presentations and 44 as posters. Since 2008 the conference is almost single track, and therefore there was no real grading in quality between oral and poster papers. As an acknowledgment that CIARP has established itself as a high-quality conference, its proceedings appear in the *Lecture Notes in Computer Science* series. Moreover, its visibility is further enhanced by a selection of a set of papers that will be published in a special issue of the journal *Pattern Recognition Letters*.

The conference program was highlighted by invited talks by four internationally leading scientists, Maria Petrou, Peter Sturm, Walter G. Kropatsch and Ioannis A. Kakadiaris, with topics on imaging architectures, 3D geometric modeling, pyramid representations and methods for analyzing CT data. Professors Petrou, Sturm and Kropatsch also contributed to the overall goal of promoting knowledge in the field in their tutorials on texture analysis, geometric methods in computer vision and pyramid representations. In two additional tutorials Eduardo Bayro-Corrochano and Dietmar Hildebrand presented insights on applying and implementing geometric algebra techniques in robot vision, graphics and medical image processing.

The full-day CASI 2009 Workshop on Computational Advances of Intelligent Processing of Remote Satellite Imagery, co-sponsored by IEEE GRSS and chaired by Yuriy Shkvarko, CINVESTAV, Unidad Guadalajara, was held in connection with the conference. For the CASI 2009 Workshop, after a double-blind review proces, 12 papers were accepted.

Another event that increaded the interest of the conference was the preceding first Mexican Workshop on Pattern Recognition (MWPR 2009). MWPR 2009 was organized by the Mexican Association for Computer Vision, Neural Computing ad Robotics (MACVNR). It was sponsored by the Computer Science Department of the National Institute of Astrophysics, Optics and Electronics (INAOE), and the Center for Computing Research of the National Polytechnic Institute (CIC-IPN). The aim of MWPR 2009 was to be a forum for exchanging scientific results and experiences, as well as sharing new knowledge, and increasing the co-operation between research groups in pattern recognition and related areas, in México.

As co-organizers of CIARP 2009, we would like to express our gratitude to both the supporting organizations and all those who contributed to the conference in other ways. We gratefully acknowledge the support from CINVESTAV and MACVNR and the other five Iberoamerican PR societies supporting the main meeting, as well as the support offered by the International Association for Pattern Recognition. We also extend our thanks to the organizations supporting our workshops.

We are particularly grateful to the Organizing Committee and the Program Committte for their devoted work leading to an impeccable review process. A special thanks must inevitably go to the members of the organizing Committee, who made this conference an excellent event through their serious work.

Finally, a conference is only as good and fruitful as the participants make it. We therefore, last but certainly not least, extend our deepest gratitude to all those who by their presence and contributions made this an excellent conference. We hope they enjoyed the meeting as much as we did.

September 2009

Eduardo Bayro-Corrochano
Jan-Olof Eklundh

Organization

The 14th Iberoamerican Congress on Pattern Recognition (Congreso Ibero-Americano de Reconocimiento de Patrones CIAP 2009) was held in Guadalajara, Jalisco, Mexico during November 15–18, 2009, and organized by the Mexican Association of Computer Vision, Neurocomputing and Robotics (MACVNR), endorsed by the International Association for Pattern Recogntion (IAPR).

General Chairs

Eduardo Bayro-Corrochano	CINVESTAV, Guadalajara, Mexico
Jan-Olof Eklundh	KTH, Stockholm, Sweden

CASI 2009 Workhop Chair

Yuriy Shkvarko	CINVESTAV, Guadalajara, Mexico

IAPR-CIARP 2009 Award Committee

Ioannis Kakadiaris	University of Houston, Texas, USA
Jan-Olof Eklundh	KTH, Stockolm, Sweden
Walter Kropatsch	TH University Viena, Austria
Maria Petru	Imperial College, London, UK

Organizing Committee

Eduardo Bayro-Corrochano	CINVESTAV, Guadalajara, Mexico
Andrez Méndez-Vázquez	CINVESTAV, Guadalajara, Mexico
Miguel Bernal-Marin	CINVESTAV, Guadalajara, Mexico
Ruben Machucho-Cadena	CINVESTAV, Guadalajara, Mexico
Heriberto Cazarubias-Vargas	CINVESTAV, Guadalajara, Mexico
Alberto Petrilli-Baserselo	CINVESTAV, Guadalajara, Mexico
Carlos López-Franco	Universidad de Guadalajara, Mexico

CIARP Steering Committee

Hector Allende	AChiRP Chile
Helder Araujo	APRP Portugal
Eduado Bayro-Coorrochano	MACVNR Mexico
Cesar Beltran Castañon	PAPR Peru
Jose Ruiz-Shulcloper	ACRP Cuba

Alberto Sanfeliu	AERFAI Spain
Alvaro Pardo	APRU Uruguay
Hemerson Pistori	SIGPR-SBC Brazil

Program Committee

Eduardo Bayro-Corrochano	CINVESTAV, Guadalajara, Mexico
Andrez Méndez-Vázquez	CINVESTAV, Guadalajara, Mexico
Carlos Lopez-Franco	Universidad de Guadalajara, Mexico
Miguel Bernal-Marin	CINVESTAV, Guadalajara, Mexico
Ruben Machucho-Cadena	CINVESTAV, Guadalajara, Mexico
Jaime Ortegon	Universidad de Quintana-Roo, Mexico
Jorge Rivera-Rovelo	Universidad de Anahuac Mayab, Mexico

Sponsoring Institutions

International Association for Pattern Recogntion (IAPR)
Mexican Association for Computer Vision, Neurocomputing and Robotics (MACVNR)
Cuban Association for Pattern Recogntion (ACRP)
Chilean Association for Pattern Recogntion (AChiRP)
Special Interest Group of the Brazilian Computer Society (SIGPR-SBC)
Spanish Association for Pattern Recogntion and Image Analysis (AERFAI)
Portuguese Association for Pattern Recogntion (APRP)
CINVESTAV, Unidad Guadalajara, Jalsico, México
IEEE GRSS
CoecytJal, Jalisco, México
INTEL Education
Dirección de Turismo Guadalajara, Gobierno Municipal
Oficina de Visitantes y Convenciones de Guadalajara, A.C.

Reviewers

Alquézar René	Universitat Politécnica de Catalunya, Spain
Altamirano Leopoldo	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Antonacopoulos Apostolos	University of Salford, UK
Arana Nancy	Universidad de Guadalajara, Mexico
Arias Estrada Miguel	Inst. Nacional Astrofísica, Óptica Electrónica, Mexico
Asano Akira	Hiroshima University, Japan
Bagdanov Andrew	Universidad Autónoma de Barcelona, Spain
Bayro Corrochano Eduardo	CINVESTAV, Inst. Politécnico Nac., Mexico

Begovich Ófelia	CINVESTAV - Guadalajara, Mexico
Bellon Ólga	Universidade Federal do Paraná, Brazil
Borges Dibio	Universidade de Brasilia, Brazil
Brun Luc	GREYC, France
Carrasco-Óchoa Jess A.	Inst. Nacional Astrofísica, Óptica Electrónica, Mexico
Castelán Mario	Centro de Investigación y Estudios Avanzados del I.P.N., Mexico
Castellanos Sánchez Claudio	LTI, Cinvestav - Tamaulipas, Mexico
Cheng Da-Chuan	China Medical University, Taiwan
Dawood Mohammad	University of Münster, Germany
Del Bimbo Alberto	Universita degli Studi di Firenze, Italy
Denzler Joachim	Friedrich-Schiller University of Jena, Germany
Du Buf Hans	University of Algarve, Portugal
Duin Robert P.W.	Delft University of Technology, The Netherlands
Dunham Margaret	Southern Methodist University, USA
Enrique Sucar Luis	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Escalante Ramírez Boris	Universidad Nacional Autónoma de México, Mexico
Escolano Francisco	University of Alicante, Spain
Facon Jacques	Pontifícia Univ. Católica do Paraná, Brazil
Ferri Francesc	Universidad de Valencia, Spain
Fink Gernot	Technische Universität Dortmund, Germany
Florez Mendez Alejandro	Universidad La Salle, Mexico
Foggia Pasquale	Università di Napoli Federico II, Italy
Fred Ana	Instituto Superior Técnico, Portugal
G. Zagoruiko Nikolai	Novosibirsk State Tech. Univ. (NSTU), Russia
Gómez-Ramírez Eduardo	LIDETEA, Universidad La Salle, Mexico
Garcia Mireya	CITEDI-IPN, Mexico
Gelbukh Alexander	Instituto Politécnico Nacional, Mexico
Gerardo de la Fraga Luis	Cinvestav. Department of Computing, Mexico
Ghosh Anarta	Research Fellow, Ireland
Gibert Karina	Univ. Politécnica de. Cataluña, Spain
Gomes Herman	Univ. Federal de Campina Grande, Brazil
González Jordi	Sabate Universitat Politecnica de Catalunya, Spain
Gonzalez Jesus	National Institute of Astrophysics, Óptics and Electronics, Mexico
Grana Manuel	University of the Basque Country, Spain
Grau Antoni	Universidad Politécnica de Cataluña, Spain
Grigorescu Cosmin	European Patent Óffice, The Netherlands
Haindl Michal	Czech Academy of Sciences, Czech Republic
Hanbury Allan	Vienna University of Technology, Austria

Hancock Edwin	University of York, UK
Hernando Javier	Univ. Politecnica de Catalunya, Barcelona, Spain
Heutte Laurent	Université de Rouen, France
Hlavac Vaclav	Czech Technical University, Czech Republic
Ho Tin Kam	Bell Labs, Alcatel-Lucent, USA
Huang Yung-Fa	Chaoyang University of Technology, Taiwan
Jan-Ólof Eklundh	KTH, the Royal Institute of Technology, Sweden
Jiang Xiaoyi	Universität Münster, Germany
Kakadiaris Ioannis A.	University of Houston, USA
Kampel Martin	Vienna University of Technology, Austria
Kim Sang-Woon	Myongji University, Korea
Kittler Josef	University of Surrey, UK
Klette Reinhard	The University of Auckland, New Zealand
Kober Vitaly	CICESE, Mexico
Kodratoff Yves	CNRS & Université Paris-Sud, France
Koschan Andreas	University of Tennessee Knoxville, USA
Kosters Walter	Universiteit Leiden, The Netherlands
Kropatsch Walter	Vienna University of Technology, Austria
López Aurelio	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Llados Josep	Computer Vision Center, Universitat Autonoma de Barcelona, Spain
Lopez de Ipiña Peña Miren	Escuela Universitaria Politécnica de Donostia-San Sebastián, Spain
Lopez-Arevalo Ivan	Laboratory of Information Technology, Cinvestav - Tamaulipas, Mexico
Lopez-Franco Carlos	Universidad de Guadalajara, Mexico
Lopez-Juarez Ismael	CINVESTAV - Saltillo, Mexico
Lorenzo Ginori Juan V.	Universidad Central de Las Villas, Cuba
Martínez-Trinidad José F.	Inst. Nacional Astrofísica, Óptica Electrónica, Mexico
Mascarenhas Nelson	Federal University of Sao Carlos, Brazil
Mejail Marta	University of Buenos Aires, Argentina
Miguel Benedi Jose	DSIC, UPV, Spain
Moctezuma Miguel	Universidad Nacional Autónoma de Mexico, Mexico
Montes Manuel	Computer Science Department, INAÓE, Mexico
Morales Eduardo	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Munoz-Melendez Angelica	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Murino Vittorio	University of Verona, Italy
Niemann Heinrich	University of Erlangen-Nürnberg, Germany
Novovicova Jana	Institute of Information Theory and Automation, Czech Academy of Sciences, Czech Republic

Ochoa Rodríguez Alberto	Instituto de Cibernética, Matemática y Física, Cuba
Ortegón Jaime	Universidad de Quintana Roo, Mexico
Pardo Alvaro	Inst. de Matemáticas y Estadística, Uruguay
Petkov Nicolai	University of Groningen, The Netherlands
Petrou Maria	Imperial College London, UK
Pietikainen Matti	University of Óulu, Finland
Pina Pedro	Faculdad de Ciencias, Univ. de Porto, Portugal
Pinho Armando	DETI / IEETA Univ. de Aveiro, Portugal
Pinto Joao	Instituto Superior Tecnico, Portugal
Pistori Hemerson	Universidade Católica Dom Bosco, Brazil
Pizlo Zygmunt	Purdue University, USA
Pla Filiberto	Universitat Jaime Castelló, Spain
Ponomaryov Volodymyr	National Polytechnic Inst. of Mexico, Mexico
Pons-Porrata Aurora	Universidad de Óriente Santiago de Cuba, Cuba
Radeva Petia	Universitat Autonoma de Barcelona, Spain
Ramírez-Torres Gabriel	Centro de Investigacion y Estudios Avanzados, Mexico
Randall Gregory	Universidad de la Republica, Uruguay
Real Pedro	Universidad de Sevilla, Spain
Reyes-Garcia Carlos A.	Inst. Nac. Astronomía, Óptica Electrónica, Mexico
Rivera Rovelo Jorge	Universidad Anáhuac Mayab , Mexico
Ross Arun	West Virginia University, USA
Rueda Luis	University of Concepcion, Chile
Ruiz del Solar Javier	Universidad de Chile, Chile
Ruiz-Shulcloper Jose	Advanced Technologies Applications Center (CENATAV) MINBAS, Cuba
Sablatnig Robert	Vienna University of Technology, Austria
Sanches João	Universidade Tecnica de Lisboa, Portugal
Sanniti di Baja Gabriella	Institute of Cybernetics E.Caianiello, CNR, Italy
Sansone Carlo	Universita di Napoli Federico II, Italy
Shirai Yoshiaki	Ósaka Univ. at Suita -Ritsumeikan Univ., Japan
Shkvarko Yuriy	CINVESTAV, Mexico
Sossa Azuela Humberto	National Polytechnic Institute, Mexico
Stathaki Tania	Imperial College London, UK
Sturm Peter	INRIA, France
Sugimoto Akihiro	National Institute of Informatics, Japan
Taboada Crispi Alberto	Univ. Central Marta Abreu de Las Villas, Cuba
Tao Dacheng	The Hong Kong Polytechnic University, Hong Kong
Tombre Karl	Inst. Nat. Polytechnique de Lorraine, France
Torres-Méndez Luz Abril	CINVESTAV Unidad Saltillo, Mexico
Valev Ventzeslav	Saint Louis University, USA
Vallejo Aguilar J. Refugio	Universidad de Guanajuato, Mexico

Vilasis Xavier	Universitat Ramon Llull, Barcelona
Wang Shengrui	University of Sherbrooke, Quebec, Canada
Westenberg Michel	Eindhoven University of Technology, The Netherlands
Whelan Paul F.	Dublin City University, Ireland
Zamora Julio	Intel Research Center, Mexico
Zhou Zhi-Hua	Nanjing University, China

Table of Contents

I Keynote 1

II Image Coding, Processing and Analysis

III Segmentation, Analysis of Shape and Texture

IV Keynote 2

V Geometric Image Processing and Analysis

VI Analysis of Signal, Speech and Language

VII Document Processing and Recognition

VIII Keynote 3

IX Feature Extraction, Clustering and Classification

X Statistical Pattern Recognition

XI Neural Networks for Pattern Recognition

XII Keynote 4

XIII Computer Vision

XIV Video Segmentation and Tracking

XV Robot Vision

XVI Keynote 5

XVII Intelligent Remote Sensing Imagery Research and Discovery Techniques

XVIII CASI 2009 Workshop I: Intelligent Computing for Remote Sensing Imagery

XIX CASI 2009 Workshop II: Intelligent Fussion and Classification Techniques

I Keynote 1

An Imaging Architecture Based on Derivative Estimation Sensors

Maria Petrou and Flore Faille

Imperial College London, SW7 2AZ, UK

Abstract. An imaging architecture is proposed, where the first and second derivatives of the image are directly computed from the scene. Such an architecture bypasses the problems of estimating derivatives from sampled and digitised data. It, therefore, allows one to perform more accurate image processing and create more detailed image representations than conventional imaging. This paper examines the feasibility of such an architecture from the hardware point of view.

1 Introduction

In many image processing and computer vision operations we have to use the brightness derivatives of the observed scene. Examples of such operations include all methods that rely on gradient estimation, Laplacian estimation or estimation of higher order fluctuations. Applications include edge detection, multiresolution image representation using the Laplacian pyramid, all methods that rely on anisotropic (or isotropic) diffusion and even wavelet-based methods. In all these cases the derivatives needed are calculated from the discrete data. Discretisation, however, introduces significant errors in the calculation of differentials. An example is shown in figure 1, taken from [26], where the discrete and the continuous wavelet transform of a signal are shown. We can easily appreciate how gross a frequency representation the wavelet transform computed from the sampled version of the signal is.

The importance of processing in the analogue domain has become more evident in the recent years. Splines [28] is a very versatile and powerful tool for representing the discrete data in the continuous domain. Joshi [24] has shown that much improved histogram estimates of the data may be obtained by upsampling and interpolating the data before calculating the histograms. There have also been cases where people try to go back to the continuous domain by emulating "continuous" sensors. In [27] virtual cameras were introduced, with spectral responses in between the discrete spectral responses of actual cameras, in order to improve colour segmentation. In [23] virtual sensors measuring the potential at border points of a 2D vector field were introduced in order to improve the vector field reconstruction using the inverse Radon transform.

It is not necessary, however, for the extra sensors introduced to measure the same quantity as the existing sensors. It is true, that a much denser array of CCDs will sample the brightness of the scene much better than a not so dense array, and it will make the estimated derivatives approach more the true ones

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 3–18, 2009.

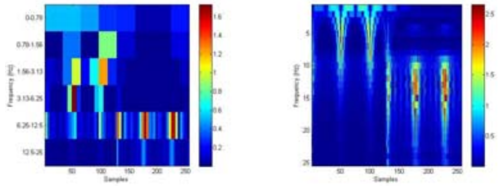


Fig. 1. The discrete (left) and continuous (right) wavelet transform of a signal

that refer to the continuous scene. There is another, option, however: it may be possible to use extra sensors that measure the desired quantities directly from the continuous scene. We started by mentioning the significant role differentiation plays in image processing. We would suggest that we may incorporate sensors in our imaging devices that measure the first and second derivatives of the scene directly, as they measure the brightness of the scene. This may be done densely and for different colour bands. The information from the derivative sensors may be used for subsequent processing, as extra information alongside the brightness information, for example in the form of extra image bands, or it may be used to construct a very accurate representation of the scene, much more accurate than a single layer of brightness sensors may do on their own. This interlacing of sensors of different types and different sensitivities appears to sound too complicated, but it may be what nature has implemented for us. In [25], a connection of such a scheme to the architecture of the human retina was made. Some aspects of the retina structure could be explained by the scheme and some predictions were made concerning the photosensitivity of the retinal cells.

In this paper, we attempt to answer the question concerning the feasibility of such an imaging system, from the hardware point of view. We present an overview of existing hardware systems for estimating first and second order derivatives directly from continuous signals. This paper is organised as follows. In Section 2 we present the theoretical underpinnings of the proposed scheme. In Section 3 we consider how such a device might be realised in hardware. We conclude in Section 4.

2 Theoretical Considerations

Reconstructing an image from its derivatives requires the integration of the field of derivative values. In this section we consider the process of integration in the continuous and the digital domain and the role the constants of integration play in the process.

2.1 Constants of Integration

Assume that we know the second derivative $d^2f(x)/dx^2$ of a function $f(x)$. What are the values of function $f(x)$? We have to integrate $d^2f(x)/dx^2$ twice: first we find the first derivative of the function

$$\frac{df(x)}{dx} = \int \frac{d^2f(x)}{dx^2}dx + c_1 \tag{1}$$

where c_1 is a constant of integration. Then we have to integrate $df(x)/dx$ once more to derive function $f(x)$

$$\begin{aligned} f(x) &= \int \left(\int \frac{d^2 f(x)}{dx^2} dx + c_1 \right) dx + c_2 \\ &= \int \left(\int \frac{d^2 f(x)}{dx^2} dx \right) dx + c_1 x + c_2 \end{aligned} \tag{2}$$

where c_2 is another constant of integration.

Note that the constants of integration appear because we perform an indefinite integral. When we perform definite integrals between pre-specified lower and upper limits, say a and b, the result we get is a numerical value of the area under the curve of the integrand between these two limits.

Now, let us consider digital integration. In the digital domain, differentiation is replaced by differencing and integration by summation. The summation, however, is between specific values of the summation index, and so it really corresponds to the *definite* integration of the continuous domain. What in the digital domain corresponds to the *indefinite* integration of the continuous domain is the recovery of the values of the differenced function at all sample positions, by propagating a starting value. We shall explain this with a specific example.

Assume that the true values of a function in a succession of sampling points are:

$$x_1, \quad x_2, \quad x_3, \quad x_4, \ldots, x_N \tag{3}$$

Assume that we are given only the first difference values at each of the sampling points, defined as $d_i \equiv x_i - x_{i-1}$:

$$\begin{aligned} &?, d_2, d_3, d_4, \ldots, d_N \equiv \\ &?, x_2 - x_1, x_3 - x_2, x_4 - x_3, \ldots, x_N - x_{N-1} \end{aligned} \tag{4}$$

Here the question mark means that we do not have the value at the first point due to the definition we used for d_i. To recover the values of the original sequence, from the knowledge of the d values, we hypothesise that the first value of the sequence is c_1. Then, the recovered values are:

$$\begin{aligned} &c_1, \quad c_1 + d_1, \quad c_1 + d_1 + d_2, \quad c_1 + d_1 + d_2 + d_3, \\ &c_1 + d_1 + d_2 + d_3 + d_4, \ldots, c_1 + d_1 + d_2 + \ldots + d_N \end{aligned} \tag{5}$$

This process corresponds to the indefinite integration of the continuous case, with constant of integration the guessed original value c_1.

There are three important observations to make.

- Without the knowledge of c_1 it is impossible to reconstruct the sequence.
- To recover the value at a single point we need to add the values of several input points.
- As the sequence is built sample by sample, any error in any of the samples is carried forward and is accumulated to the subsequent samples, so the Nth sample will be the one with the most erroneous value.

There are two conclusions that can be drawn from the above observations.

- Such reconstructions cannot be too long, as very quickly the error of reconstruction accumulates and the reconstruction becomes useless. So, for the reconstruction of a long sequence, one has to consider many small sequences in succession, and possibly with overlapping parts.
- If one has a series of sensors that return the local difference value of the observed scene, one needs another series of sensors that return the value of c_1 every so often in the sequence, ie at the beginning of every small reconstruction sequence.

Next, assume that the array of sensors we have does not measure the first difference of the sequence, but the second difference, $dd_i \equiv d_i - d_{i-1}$. Then we must apply the above process of reconstruction once in order to get the sequence d_i and then once more to get the x_i values. Note that this implies that we must have a series of sensors that every so often in the long sequence of dd_i will supply the starting constant we need, which in this case is denoted by c_2. This constant is actually a first difference, so these sensors should measure the first difference at several locations.

2.2 The Basic Idea of the Imaging Device in 1D

A device that functions according to the principles discussed above, has to consist of five layers, as shown in figure 2.

The function of this structure effectively repeats twice: below the dashed line we have the first integration, outputting above the dashed line the values of the first difference it computes, and above the dashed line we have the second integration, integrating the first differences it receives and outputting the signal values.

2.3 Extension to 2D

The analysis done in the previous two sections is in 1D. However, images are 2D. This has some serious implications, particularly for the c_2 sensors.

From the mathematical point of view, once we move to 2D, we are dealing with 2D integrals, not 1D. A 2D integration implies spatially dependant constants of integration. For a start, a 2D function $f(x, y)$ has two spatial derivatives, $\partial f/\partial x$ and $\partial f/\partial y$. Let us assume that we know both of them and we wish to recover function $f(x, y)$ by integration. Integrating the first one of them will yield

$$f(x, y) = \int \frac{\partial f}{\partial x} dx + c_x(y) \qquad (6)$$

where $c_x(y)$ is a function of y, which, as far as integration over x is concerned, is a constant. Differentiating result (6) with respect to y should yield $\partial f/\partial y$, which is known, and this can help us work out constant $c_x(y)$ as a function of y.

There is an alternative route to work out $f(x, y)$. Integrating the partial derivative with respect to y we get

$$f(x,y) = \int \frac{\partial f}{\partial y} dy + c_y(x) \tag{7}$$

where $c_y(x)$ is a function of x, which as far as integration over y is concerned, is a constant. Differentiating result (7) with respect to x should yield $\partial f/\partial x$, which is known, and this can help us work out constant $c_y(x)$ as a function of x.

Obviously, both routes should yield the same answer. In the digital domain, this corresponds to reconstruction of the 2D signal either line by line or column by column. So, let us assume that the true values of the 2D digital signal are g_{ij}. However, we do not have these values, but we are given instead the first differences of the digital signal along both directions. So, we assume that we have $dx_{ij} \equiv g_{ij} - g_{i-1,j}$ and $dy_{ij} \equiv g_{ij} - g_{i,j-1}$. We can construct the signal column by column as follows. First column:

$$\begin{aligned} g_{12} &= dy_{12} + c_y(1) \\ g_{13} &= dy_{13} + dy_{12} + c_y(1) \\ &\ldots \end{aligned}$$

Second column:

$$\begin{aligned} g_{22} &= dy_{22} + c_y(2) \\ g_{23} &= dy_{23} + dy_{22} + c_y(2) \\ &\ldots \end{aligned}$$

And similarly for the rest of the columns. This is shown in figure 3a. In a similar way, the signal may be reconstructed along rows. First row:

$$\begin{aligned} g_{21} &= dx_{21} + c_x(1) \\ g_{31} &= dx_{31} + dx_{21} + c_x(1) \\ &\ldots \end{aligned}$$

Second row:

$$\begin{aligned} g_{22} &= dx_{22} + c_x(2) \\ g_{32} &= dx_{32} + dx_{22} + c_x(2) \\ &\ldots \end{aligned}$$

And similarly for the rest of the rows. This is shown in figure 3b.

Of course, these reconstructions should be equivalent, ie one expects that $g_{22} = dy_{22} + c_y(2) = dx_{22} + c_x(2)$. One may also reconstruct the signal by using a combination of rows and columns, and again, the reconstruction should be the same irrespective of the path followed. This is shown in figure 3c.

There are two problems with the above analysis: in practise the alternative reconstructions are never identical due to noise. This is something well known

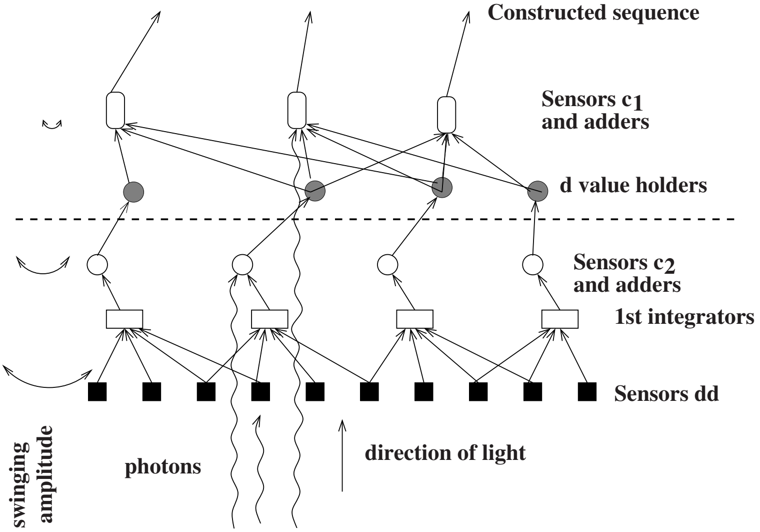


Fig. 2. Sensors *dd* measure the second derivative, while sensors c_2 the first derivative (the constant of integration for the first integration) and sensors c_1 the value of the function (the constant of integration for the second integration)

from digital image processing. The other problem is the use of two directions which creates an anisotropic grid, as there are two preferred orientations. Along these two orientations, the samples used are at a fixed distance from each other. However, if we consider samples that are aligned along the diagonal of these two orientations, their distance is $\sqrt{2}$ times that of the samples along the preferred orientations. This anisotropy is not desirable.

2.4 The Basic Idea of the Proposed Architecture in 2D

The above approach is compatible with the conventional CCD sensors that consist of rectangular cells, ie rectangular pixels. The cones in the fovea region of the retina, however, have a hexagonal structure, as shown in figure 4a. At first sight this does not look very useful. However, instead of considering the cells, consider their centres as the sampling points of a grid. The nodes in the grid shown in figure 4b are the points where the reconstruction has to take place.

This sampling grid at first sight does not appear hexagonal, but rather based on equilateral triangles. However, several hexagons of various scales can be perceived here.

Imagine now, that we have a device centred at the centre of one of these hexagons. Imagine that the device vibrates along the paths shown. Imagine that this device hangs from a vertical nail above the centre of the hexagon, and consists of three types of sensor hanging from the same string: the bottom one measures second differences, the middle one first differences, and the top one just values. As the string swings like a pendulum, the bottom sensor swings

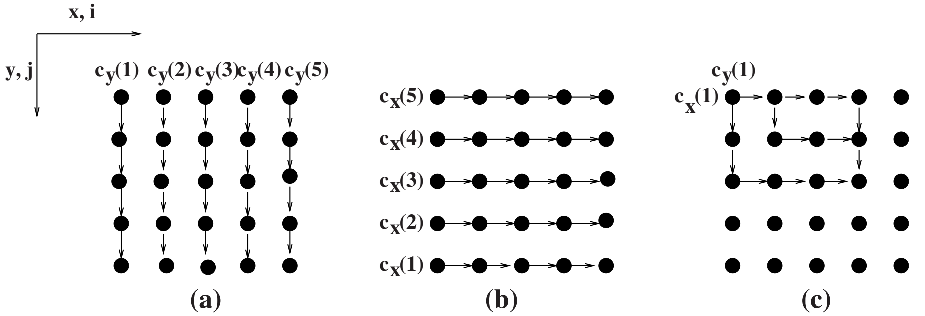


Fig. 3. Reconstruction from first difference values in 2D can proceed along columns (a), or rows (b), or along any path (c). The answers should be equivalent.

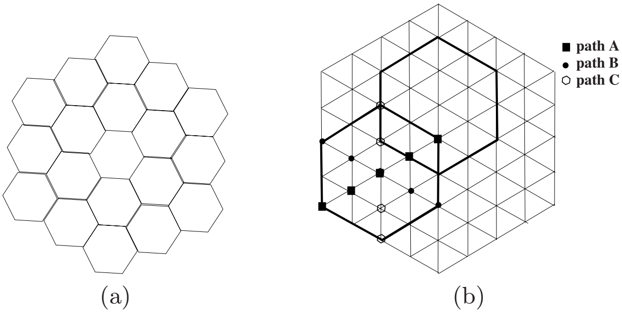


Fig. 4. (a) The arrangement of cells in the mammalian retina. (b) The hexagonal sampling grid.

more, the middle less and the top not at all (see left of figure 2). This will be consistent with the notion that the second difference sensor needs to see larger part of the scene to do its job than the first difference sensor, while the fixed sensor does not need to swing at all to do its job. Note: it is mathematically impossible to calculate any derivative if you consider only a single sample. So, a device like the one shown in figure 2 swinging along one direction, will allow the reconstruction of the signal along that direction for several sampling points. The amplitude of the swing and the range of reconstruction performed by each single set of sensors are two different things. The amplitude of the swing is for measuring locally what is needed for the reconstruction. Swinging along another direction, will measure the first and second differences along that direction, and the signal will be reconstructed along that direction, by using the propagation techniques we discussed in the 1D case.

There are many advantages of this approach: the reconstruction grid is isotropic; we have no preferred directions; the hexagons fit nicely with each other at all scales; the reconstruction along the lines of one hexagon can be complemented by the reconstruction along the lines of other hexagons that may be directly underneath other sets of sensors hanging from our swinging strings; overlapping reconstructions are expected to add robustness and super-acuity (ie resolution higher than the sampling distance as determined by the spacing of the sensors); the reconstruction is expected to be complete and very accurate.

3 Hardware Considerations

The proposed imaging framework consists of three types of sensor: brightness sensors and sensors that measure spatial first and second order derivatives. These spatial derivatives are obtained by moving the photosensors and computing temporal first and second order derivatives. The photosensor motion is fast and of small amplitude. This is inspired from the microsaccades in the human visual system. After the data acquisition, the image may be reconstructed by using the values of the derivatives, as well as the brightness values.

In a hardware realisation of such an imaging device, brightness values would be acquired with photodiodes, which transform the energy received by photons into electrical charges. Detector motion could be realised by moving the optic or a masking element in front of the photoreceptors, using microsized actuators, like the piezoelectric actuator presented in the micro–scanning device in [8]. As far as imaging technology is concerned, CMOS should be used because charges are transformed into voltages or currents directly at the pixel. This allows one to start the processing (the derivative estimation in our case) before any read–out is performed. However, CMOS technology suffers from higher noise sensitivity than CCD cameras. Typical problems are fixed pattern noise, which results from transistor mismatches on the chip, and higher noise sensitivity in dark lighting conditions when the chip has a low fillfactor (ratio of the photosensitive chip area over the total chip area). Image quality is typically improved using processing like the double sampling technique against fixed pattern noise and by keeping the fillfactor as high as possible [16].

In our application, brightness values, as well as first and second order derivatives must be measured with a good precision in order to achieve good image reconstruction quality. Brightness sensors can be implemented like typical pixels in a CMOS camera, with double sampling to reduce fixed pattern noise (see e.g. [16]). For the first and second order derivative sensors, the photodiode signal must be processed to estimate the temporal derivatives. To keep the fillfactor high and to allow a high density of derivative sensors, the circuit for derivative estimation should be as small as possible if processing is performed in situ. Derivatives will not be affected by fixed pattern noise.

Three possible technologies can be used to estimate temporal derivatives from the photodiode signals: fully digital processing, discrete time analog processing and fully analog processing. Digital processing offers the best robustness against electronic noise but it requires a large circuit size and the power consumption is high. On the contrary, a fully analog solution allows a small and energy–efficient circuit, at the cost of a higher sensitivity to electronic noise and parasitic effects caused e.g. by transistor mismatches. Discrete time analog processing offers an intermediate solution: photodiode values are sampled at regular time intervals but they are not digitised. Processing is performed with analog circuits. The three subsections that follow describe existing systems that estimate derivatives using one of these three technologies. Most of these systems deal with imaging sensors or audio applications. In particular, many VLSI based motion sensors can be found, because the high volume of the data generated by such applications can be handled more easily by parallel hardware, like a VLSI chip, than by standard serial processors. We could not find any existing systems computing temporal second order derivatives. However, they can be built similarly to the first order derivatives for fully digital processing and for discrete time analog processing. For analog processing, they can be obtained by putting two first order derivative circuits in series.

3.1 Digital Circuits

In fully digital circuits, signals are sampled in time and digitised during acquisition. As a result, signals are affected by aliasing and discretisation noise. Digital circuits are however a lot less sensitive to electric noise and distortions because values are encoded as binary and because transistors are only used in their saturated and blocked states. Digital circuits have a high power consumption and are most of the time large circuits. Digital processing is the method of choice when data acquisition and processing can be performed at different time instants and on different systems (e.g. the processing is performed on an external processor).

Here, however, processing (derivative estimation) should be performed on the chip, if possible near the photodiode. Thanks to the constant size reduction of the electronics (Moore's law), the first intelligent cameras (also named artificial retinas) with in situ digital processing have become a reality. One such system is described in [20]. Each pixel is composed of a photodiode, a simple analog to digital converter and a tiny digital programmable processing element. It comprises about 50 transistors. The processing element has a local memory and can perform boolean operations. The chip has a fillfactor of 30%. Due to the limited number of transistors, brightness values can only be encoded with few grey levels (8 grey levels in the examples given in the paper), resulting in low precision. A few simple applications have been implemented on the system: motion detection, segmentation and shape recognition. However, our application requires high precision. Even though the control signals, necessary for programmability, can be avoided in our case, it would be difficult to obtain the necessary precision in the circuit space available near each pixel. Therefore, fully digital processing seems not appropriate for our application at the moment, except if derivative computations are performed externally (on a separate processor). However, this defeats the purpose of our approach, which is the direct and in situ estimation of the derivatives.

3.2 Discrete Time Circuits

For these systems, the photodiode signals are sampled at given time instants and they are stored in local analog memories or sample-and-hold units. Processing is performed using the stored pixel values on analog circuits. Like for the fully digital circuits, the system operates in two phases: sampling, followed by processing. Such circuits represent an intermediate solution between fully analog and fully digital systems. Like digital systems, they might be affected by temporal aliasing (which corresponds to spatial aliasing in our case, because spatial derivatives are estimated using sensor motion and temporal derivatives). However, brightness values are not digitised, so there is no discretisation noise. Control signals for the sampling should be generated and synchronised to the sensor motion. Like in analog systems, computations are less robust to electronic noise and distortions than in fully digital systems. Noise and distortions are caused, for example, by active filter elements, like amplifiers, by transistor mismatches and other element inaccuracies, by parasitic capacitors, by leakage currents, etc.

Discrete time intelligent camera systems started being designed when CMOS cameras became popular. Early systems only added local memories to pixels (analog memories or sample-and-hold units, composed, for example, of two transistors, used as switches, and a capacitor to store the value [16]). Processing was performed during the image read out phase, just before analog to digital conversion [6,16]. This is used in [6] to perform spatial convolutions with 5 × 5 kernels with coefficients in $\{-1, 0, 1\}$. In [16], the designed camera can deliver either normal images or differences between consecutive frames. More recent systems, based on the same organisation (separated photosensitive and processing areas), can perform more elaborate processing, like optical flow computation in [18] or saliency detection in [9]. The separation of photosensitive and processing areas allows high fillfactors (e.g. 40% in [9]). However, processing is performed row by row, which limits the achievable frame rate. These last two systems [18,9] compute temporal derivatives by subtracting pixel values at two time instants, t and $t + \Delta t$: $d(t) = I(t) - I(t - \Delta t)$, where Δt is different from the sampling period of the camera. In [18], a calibration scheme is included to suppress the distortions caused by mismatches between the p and n elements.

A more recent system, where temporal derivatives are estimated by differentiating the sensor values at two different time instants, is given in [21]. Sample-and-hold units are implemented with switched capacitor technology, which allows high accuracy and programmability of the capacitances. In addition to differentiation, sample values are also amplified with a simple inverting amplifier. This system implements an audio application, with which sound sources can be localised. Space is therefore not an issue for them, unlike for imaging systems. A recent intelligent camera is presented in [4], in which processing is performed in situ. Each pixel is composed of a photodiode and 38 transistors. It contains two analog memories (to allow the acquisition of the next image during the processing of the current image) and a simple analog arithmetic unit (to perform spatial convolution with integer based kernels). As a result of this organisation (in situ processing), a high framerate of several thousands of images per second can be achieved even when image processing is performed. The fillfactor is 25%.

This last example shows that analog computation of derivatives could be performed in situ if a discrete time circuit is chosen for the implementation of our imaging sensor. None of the papers gives any indication about noise level or whether distortions influence the precision of the results. However, the time discretisation allows one to reduce the circuit complexity: derivation is replaced by a simple subtraction of two sampled values. The results should therefore be quite accurate. Second order derivatives can be implemented similarly by sampling data at three time instants and by performing two additions and one subtraction: $dd(t - \Delta t) = I(t - 2\Delta t) + I(t) - (I(t - \Delta t) + I(t - \Delta t))$. The signals required to control the sample-and-hold units could be generated from the signals used to control the actuators, therefore synchronising sampling and sensor motion.

3.3 Analog Circuits

In fully analog systems, analog signals are processed in continuous time. The circuit usually consumes low power and is of small size. However, analog processing is the most sensitive to electronic noise and distortions, which are caused by active elements, like amplifiers, element inaccuracies, transistor mismatches, frequency response of the circuits, parasitic capacitances, current leakage, etc. The difference between idealised model and practice is the biggest for analog circuits due to the complexity of the underlying physical phenomena. Analog circuits have the advantage of not being limited by the camera frame rate, as they are working in continuous time. Therefore they are not affected by aliasing or discretisation noise. The classical method of computing derivatives in an analog circuit is through the use of a capacitor for which $I = C\frac{dV}{dt}$. However, an ideal differentiator would require high power for high frequency signals, as the ideal transfer function of a capacitor is:

$$H(j\omega) = \frac{I_{out}}{V_{in}} = j\omega C. \tag{8}$$

This is physically unrealisable and in a real system, some resistance will always limit the current. This results in the well–known RC filter shown in figure 5. The derivative of a signal can be measured as the voltage through the resistor. The transfer function for this system is:

$$H(j\omega) = \frac{V_{out}}{V_{in}} = \frac{jRC\omega}{1 + jRC\omega} = \frac{j\tau\omega}{1 + j\tau\omega}. \tag{9}$$

For low frequencies ($\omega \ll 1/\tau$), the RC filter is a good approximation of the ideal differentiator: $H(j\omega) \approx j\tau\omega$. For high frequencies ($\omega \gg 1/\tau$), the transfer function becomes approximately 1. This simple RC filter is the basis of all analog circuits used to estimate derivatives.

In practice, amplification may be required or active resistors (i.e. non–linear resistance circuits based on transistors) may be necessary to obtain the desirable time constant $\tau = RC$ with the manufacturable circuit elements (see e.g. [19,17]). Therefore, real circuits are more complicated than a simple RC filter. Many applications do not require an accurate estimation of the derivatives. Therefore,

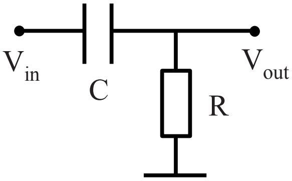


Fig. 5. RC filter used to estimate first order derivatives

many systems only detect changes in the input signal using high pass filters which are easier to realise than a good differentiator. This solution is used for example in [17,2,1,19,5]. In [17], it is shown that such an approximation can increase the robustness of a circuit (see e.g. the developed hysteretic differentiator). This approach is enough for applications aiming only at detecting motion in visual input. It is, however, not appropriate in our case, because precise derivatives are necessary for image reconstruction.

Another approach to estimate the current going through a capacitor using transistors is proposed in [12]. Instead of measuring the voltage through a resistor, the current flowing through a capacitor is copied using transistors. The photosensitive circuit and the temporal derivative circuit proposed in [12] have been extended and used in applications by the authors in [10,13] and by others in [7,3]. The temporal derivative circuit contains an amplifier and a feedback loop with a capacitor and the circuit for measuring the current going through it. As the current through the capacitor is measured, the output is, in theory, proportional to the derivatives of the light energy hitting the photodiode, as shown in [13]. The system delivers two currents: one representing negative changes in the input and one representing positive changes in the input. It is based on a functional model of the auditory hair cells that sense the motion of the basilar membrane in the cochlea. The results are quite noisy, mainly due to the amplifier [13,3]. The non–linearities and inaccuracies in the transistors also cause an asymmetric response: the signal representing negative changes reacts slightly differently from the signal representing positive changes for changes of the same amplitude [7]. Despite the noise level, the results allow one to estimate velocity in 1D in [10,13], in 2D [7] and to compute optical flow at each pixel in [3].

The same principle (measuring the current through the capacitor using transistors) is used in [14,11,15,22]. In these more recent papers, much simpler circuits are used. In particular, the amplifier is realised with fewer transistors, probably because less gain is necessary or because the transistors are of better quality. The simplest circuits are used in [14,15]. These two applications do not aim at obtaining precise derivatives but only at detecting changes in the input signal. The authors hence did not pay much attention to noise and distortions during circuit design. In [14] the resulting noise level is too high even for their application, so the authors conclude that their differentiator circuit must be improved. Therefore, the circuits in [14,15] are not appropriate for our application. The circuits in [11,22] are designed to estimate the temporal derivative of the photodiode signal as accurately as possible. Both systems use the same circuit element to read out the current through a capacitor, but the feedback loop to the amplifier is designed slightly differently in the two circuits. The results in [22] seem to be more symmetrical (the responses for positive and negative changes are more similar). However, this might be due to the fact that the article shows fewer results. The results in [11] are of much better quality than the previous papers [13,3]: the noise level is significantly reduced. However, the output signals are still far from ideal. In particular, the asymmetry between outputs for positive and for negative changes is problematic. None of the papers gives any estimation of the noise level, probably because it is influenced by many factors. Both systems in [11,22] can be implemented with one capacitor and less than 10 transistors, which is a very small circuit size.

3.4 Discussion

Three different kinds of system can be used to estimate first and second order temporal derivatives from the output of a photodiode: fully digital systems, discrete time analog systems and fully analog systems. Here we gave an overview of existing systems that estimate derivatives using any of these methods. The goal is to find a system which would be suitable to use in order to implement the new derivative based imaging sensor in hardware.

Fully digital systems offer the best robustness to electronic noise and distortions at the cost of high power consumption and large circuit areas. Only one fully digital imaging system with in situ processing could be found in the literature. It had a very limited number of grey values, which results in low precision. As a result, a fully digital in situ derivative estimation cannot be realised with today's technology. So, fully digital systems could only be used if processing would be performed on an external processor like a DSP system.

Fully analog systems have a low power consumption and more importantly they can be implemented with few transistors (in [11,22] first order derivatives are estimated with one capacitor and less than 10 transistors). On the other hand, the signals are sensitive to electronic noise and distortions caused by the non–linearities and parasitic effects in the circuits. The circuits in [11,22] allow in theory to estimate the first order derivatives accurately. The results shown are encouraging in comparison with previous works. However, they are still far from accurate. In addition to the moderate noise level, the circuit responses to positive and to negative input changes of the same amplitude are slightly different. It is therefore not certain whether a fully analog system would be accurate enough to allow a good image reconstruction. Another problem is that no circuit could be found to estimate second order temporal derivatives. These could be estimated by putting two differentiators in series, but this would also amplify noise and distortions, reducing even more the accuracy of the estimated second order derivatives.

Discrete time analog systems represent an intermediate solution. The photodiode signals are sampled at given time instants and stored in an analog memory or sample-and-hold unit. The first and second order derivatives can be estimated by subtracting the stored values. These operations are performed with analog circuits.

The resulting analog circuit is much simpler than a fully analog system, reducing problems like electronic noise and distortions. Such systems have been used recently to develop intelligent cameras (or artificial retinas) which perform not only image acquisition but simple image processing operations as well. The system in [4] shows that as many as 38 transistors can be integrated in each pixel for processing, while keeping the fillfactor at a reasonable value. This would be enough to allow the computations of first or second order derivatives. The signals used to control the actuators could be used to synchronise data sampling and sensor motion. As a result, we conclude that a discrete time analog system would be the most appropriate to be used in order to implement the proposed imaging system.

4 Conclusions

The proposed imaging architecture has several advantages over conventional architectures that measure only scene brightness:
(i) it allows the direct and accurate estimate of the first and second image derivatives directly from the scene;
(ii) it allows the increase of sensor resolution if the image is upsampled with the use of its local derivative values.
The viability of such a device rests on two fundamental questions.
1) Can we develop sensors that can estimate the first and second derivatives directly from the scene? In this paper we reviewed the current technology and concluded that discrete time analog systems are a promising direction for developing such a device. There are already sensors that can estimate the first spatial derivative of the scene, and although there are no sensors that can estimate the second spatial derivative, we do not think that such a development is too difficult or beyond the state of the art of current sensor technology.
2) Will the outputs of these sensors be more accurate and resilient to noise than the calculations of the derivatives from the sampled data? This question cannot be answered until such a device has actually been realised in hardware.
Actually, both the above questions have to be answered by sensor scientists, as they cannot be answered theoretically. There is no doubt that if the answer is “yes” to both these questions, the image processing that we shall be able to do with such devices will be much more reliable and accurate than the image processing we are doing now.

Acknowledgements. This work was supported by a UK Research Councils translation grant.

References

1. Chong, C.P., Salama, C.A.T., Smith, K.C.: Image-motion detection using analog VLSI. IEEE Journal of Solid-State Circuits 27(1), 93–96 (1992)
2. Delbrück, T., Mead, C.A.: Time-derivative adaptive silicon photoreceptor array. In: Proceedings SPIE, vol. 1541, pp. 92–99 (1991)

3. Deutschmann, R.A., Koch, C.: An analog VLSI velocity sensor using the gradient method. In: Proceedings of the 1998 International Symposium on Circuits and Systems (ISCAS 1998), pp. VI 649–VI 652 (1998)
4. Dubois, J., Ginhac, D., Paindavoine, M.: Design of a 10000 frames/s CMOS sensor with in situ image processing. In: Proceedings of the 2nd International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoc 2006), pp. 177–182 (2006)
5. Etienne-Cummings, R., Van der Spiegel, J., Mueller, P.: A focal plane visual motion measurement sensor. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 44(1), 55–66 (1997)
6. Funatsu, E., Nitta, Y., Kyuma, K.: A 128 x 128-pixel artificial retina LSI with two-dimensional filtering functions. Japanese Journal of Applied Physics 38(8B), L938–L940 (1999)
7. Higgins, C.M., Deutschmann, R.A., Koch, C.: Pulse-based 2-D motion sensors. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 46(6), 677–687 (1999)
8. Hoshino, K., Mura, F., Shimoyama, I.: Design and performance of a micro-sized biomorphic compound eye with a scanning retina. Journal of Microelectromechanical Systems 9(1), 32–37 (2000)
9. Kimura, H., Shibata, T.: A motion-based analog VLSI saliency detector using quasi-two-dimensional hardware algorithm. In: Proceedings of the 2002 International Symposium on Circuits and Systems (ISCAS 2002), pp. III 333–III 336 (2002)
10. Kramer, J.: Compact integrated motion sensor with three-pixel interaction. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(4), 455–460 (1996)
11. Kramer, J.: An integrated optical transient sensor. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 49(9), 612–628 (2002)
12. Kramer, J., Sarpeshkar, R., Koch, C.: An analog VLSI velocity sensor. In: Proceedings of the 1995 International Symposium on Circuits and Systems (ISCAS 1995), pp. 413–416 (1995)
13. Kramer, J., Sarpeshkar, R., Koch, C.: Pulse-based analog VLSI velocity sensors. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 44(2), 86–101 (1997)
14. Landolt, O., Mitros, A., Koch, C.: Visual sensor with resolution enhancement by mechanical vibrations. In: Proc. of the 2001 Conference on Advanced Research in VLSI (ARVLSI 2001), pp. 249–264 (2001)
15. Lichsteiner, P., Posch, C., Delbruck, T.: A 128x128 120dB 30mW asynchronous vision sensor that responds to relative intensity change. In: Proc. of the 2006 IEEE International Solid-State Circuits Conference (2006)
16. Ma, S.-Y., Chen, L.-G.: A single-chip CMOS APS camera with direct frame difference output. IEEE Journal of Solid-State Circuits 34(10), 1415–1418 (1999)
17. Mead, C.: Analog VLSI and Neural Systems. Addison-Wesley Publishing Company, Reading (1989)
18. Mehta, S., Etienne-Cummings, R.: Normal optical flow chip. In: Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS 2003), pp. IV 784–IV 787 (2003)
19. Moini, A., Bouzerdoum, A., Yakovleff, A., Abbott, D., Kim, O., Eshraghian, K., Bogner, R.E.: An analog implementation of early visual processing in insects. In: Proceedings of the 1993 International Symposium on VLSI Technology, Systems and Applications (VLSITSA 1993), pp. 283–287 (1993)

20. Paillet, F., Mercier, D., Bernard, T.M.: Second generation programmable artificial retina. In: Proceedings of the 12th annual IEEE International ASIC/SOC Conference, pp. 304–309 (1999)
21. Stanacevic, M., Cauwenberghs, G.: Micropower gradient flow acoustic localizer. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 52(10), 2148–2157 (2005)
22. Stocker, A.A.: Compact integrated transconductance amplifier circuit for temporal differentiation. In: Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS 2003), pp. I 25–I 28 (2003)
23. Giannakidis, A., Kotoulas, L., Petrou, M.: Improved 2D Vector Field Reconstruction using Virtual Sensors and the Radon Transform. In: International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, Canada, August 20–24 (2008)
24. Joshi, N.B.: Non-parametric probability density function estimation for medical images, PhD thesis, University of Oxford (2007)
25. Petrou, M.: A new imaging architecture and an alternative interpretation of the structure of the human retina. In: Zaman, H.B., Sembok, T.M.T., van Rijsbergen, K., Zadeh, L., Bruza, P., Shih, T., Taib, M.N. (eds.) Proceedings of the International Symposium on Information Technology, Kuala Lumpur Convention Centre, Malaysia, August 26–29, vol. 1, pp. 9–17, IEEE Cat. No CFP0833E-PRT, ISBN 978-1-4244-2327-9
26. Varnavas, A.: Signal processing methods for EEG data classification, PhD thesis, Imperial College London (2008)
27. Verges-Llahi, J.: Colour Constancy and Image Segmentation Techniques for Applications to Mobile Robotics, PhD thesis, University Politecnica de Catalunya, Barcelona, Spain (2005)
28. Unser, M.: A Guided Tour of Splines for Medical Imaging. In: Plenary talk, Twelfth Annual Meeting on Medical Image Understanding and Analysis (MIUA 2008), Dundee UK, Scotland, July 2-3 (2008)

II Image Coding, Processing and Analysis

Landmark Real-Time Recognition and Positioning for Pedestrian Navigation

Antonio Adán[1], Alberto Martín[1], Enrique Valero[1], and Pilar Merchán[2]

[1] Escuela Superior de Informática. Universidad de Castilla-La Mancha,
13071 Ciudad Real, Spain
{Antonio.Adan,Alberto.Martin,Enrique.Valero}@uclm.es
[2] Escuela de Ingenierías Industriales. Universidad de Extremadura,
06006 Badajoz, Spain
pmerchan@unex.es

Abstract. The aim of this paper is to propose a new monocular-vision strategy for real-time positioning under augmented reality conditions. This is an important aspect to be solved in augmented reality (AR) based navigation in non-controlled environments. In this case, the position and orientation of the moving observer, who usually wears a head mounted display and a camera, must be calculated as accurately as possible in real time. The method is based on analyzing the properties of the projected image of a single pattern consisting of eight small dots which belong to a circle and one dot more at the center of it. Due to the simplicity of the pattern and the low computational cost in the image processing phase, the system is capable of working under on-line requirements. This paper presents a comparison of our strategy with other pose solutions which have been applied in AR or robotic environments.

Keywords: augmented reality, camera pose, landmark, occlusion, real-time.

1 Pose through Perspective Projection Techniques

One of the key points in augmented reality systems for autonomous navigation consists of obtaining an accurate camera pose as quickly as possible. Although there are positioning and tracking systems in controlled environments - for example, technologies based on inertial devices or on networks with a signal receiver/emitter allow 3/6DOF in small environments - for autonomous systems, positioning must be solved with new solutions. Consequently, this issue continues to be an open research field in which innovative solutions are suggested every year.

Depending on each specific application and environment, several factors should be taken into account before choosing the most appropriate technique. The majority of the authors do not make any reference concerning the performance of their method when the landmark is occluded [1], [2], [3], [4], [5], [6], [7], [8]. The authors that take into account occlusion circumstances assume that the landmark is partially occluded but in a non-critical sense. Therefore, those cases correspond to non-severe occlusion. They use natural features [9], [10], [11] or artificial landmarks [12], [13], [14]. Some authors only mention that the system works under occlusion but they do not properly

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 21–28, 2009.

prove that fact ([14]) whereas others ([10], [11]) deal with the problem in depth. In [12], the designed patterns consist of a common vertical barcode and an unique horizontal barcode that distinguish it from any other. In the performed tests, all landmarks are correctly detected in despite of partial occlusions. In [13], several patterns appear in the scene. If one of them is partially occluded, pose estimation can be easily handled by using any non-occluded pattern in the image. In [15], a robust pose method analyzes the geometric distortions of the objects under changes in position. Under real-time requirements, fast image processing may be the key of the general pose approach. There are several authors who provide detailed information concerning this matter [12], [4], [15], [5], [14], [7], [10], [11]. Most of the systems argue to work in real-time conditions: specifically rates are 36-81 fps [12], 30 fps [15], [5], 10 fps [14], 7.3-8.1 fps [7], 15-25 fps [10], [11]. The size of the image, the kind of camera as well as the image processing can have an influence on the final rate of the pose system. For instance, in [4], the system works between 4.2 to 7.2 fps, depending on the kind of camera chosen. In [7], the performance of the system depends on the number and size of potential fiducials in the image.

With regard to the adaptability of the pattern being used in wide distance intervals, most of the referenced methods are designed for use indoors and they seem to work with short and constant ranges.

In any case, very little information is offered in these terms. Exceptions are: [4] where the camera itself establishes a variable range from 0.85 to 1.7 or 3.3 meters and [7] where a tracking procedure works for distances from 50 cm to 5 meters, depending on multi-ring fiducial diameter.

In this paper, the used landmark consists of eight dots corresponding to the vertices of an octahedron and one more double-size dot located at the center of the octahedron. As we will demonstrate in this paper, the system is capable of dealing with severe occlusion of the landmark. Additionally, this landmark allows us to work in a flexible range from 30 centimeters to 7 meters providing similar accuracy than that of most of the referenced approaches. Under realistic conditions, a rate of 30 fps can be achieved. There are no restrictions about the location and orientation of the landmark neither the pose of the camera. Thus the landmark can be set on the floor, ceiling, wall or wherever suitable place.

2 Parameters Definition and Pose Strategy

Before presenting the pose calculation we will outline the framework, the general pose strategy and the parameters to be calculated.

Suppose a human is wearing an AR system composed of a camera integrated into a head-mounted display (see Figure 1) and a laptop in the backpack. The reference systems to be considered are as follows: world reference system (S_w), human reference system (S_h), camera reference system (S_c), image reference system (S_e), computer reference system (S_s) (which is the digital image reference system) and landmark reference system (S_0). Note that the relationship S_w/S_0 is imposed when the landmark is positioned in a specific place and that S_h/S_c is established by the user himself. Moreover, relationship S_c/S_e and S_e/S_s are established by the intrinsic calibration of the camera. As a result, the pose problem is reduced to find the transformation S_0/S_C, which varies as the human moves.

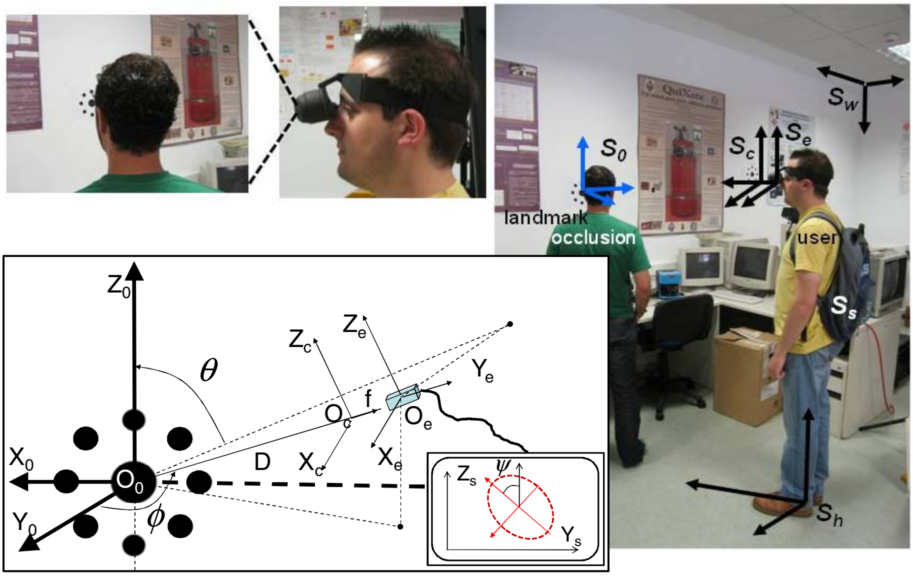


Fig. 1. Right and top) AR based navigation: Reference systems and occlusion of the landmark. Down) Parameters ψ, ϕ, θ and D' in the pattern reference system.

The autonomous procedure presented in this paper is based on the fact that, for any view of the pattern, the outer dots of the pattern belong to an ellipse *E* which changes as the person (camera) moves. Through geometric analysis of *E* and the location of the dots in it, we are able to extract the angular parameters *swing (ψ), tilt (ϕ), pan (θ)* as well as the distance *D'* between the origin of S_0 and the image plane of the camera. From this point forward, we will call the dots (or the center of the dots) P_i , i=1,2....9. (See Figure 1 to consult the pattern reference system and the parameters).

Changes in the position of the user cause changes in ellipse *E*. Thus, variation of ψ causes the rotation of the major axis of the ellipse in the image; changes in parameter ϕ imply changes in the ellipse eccentricity; when $\theta \neq 0$, dots $\{P_1, P_3, P_5, P_7,\}$ are located outside the axes of the ellipse and, finally, a variation of *D'* makes the length of the major axis of the ellipse change following a quasi-lineal relationship.

Using the camera model presented in Figure 1 down, we establish the transformation between S_0 and S_e reference systems as follows:

$$M = R_{Y_0^{''}}(\psi) \cdot T(D') \cdot R_{Y_0^{''}}(\pi/2) \cdot R_{X_0^{'}}(\phi) \cdot R_{Y_0}(-\theta) \qquad (1)$$

Where the Euler rotations are performed over axes Y_0 , $X_0^{'}$ and $Y_0^{''}$ and *T* corresponds to a translation in axis $Y_0^{''}$.

3 Pose Calculation with Occlusion

As was mentioned in section 1, occlusion circumstances frequently occur in real environments. This pattern has been designed to be used in a wide distance range (from 30 cm to 700 cm away from the user) using the same algorithm. For this reason, it is formed by small circles which in turn belong to an outer circle. Thus, for long distances, the pattern has a set of single marks belonging to a circle whereas for short distances, the pattern is seen as a set of circles. Occlusion is dealt with in this manner.

The pose algorithm with occlusion is established depending on the number of missing dots. We distinguish between several levels of occlusion.

Level 1. When one or two outer dots of the pattern are missing in the image, we categorize it as soft occlusion.

Assume that $(x_{s,i} z_{s,i})$ are the coordinates of the dots viewed in the image and that coefficients C_1, C_2, C_3, C_4, C_5 (taking coefficient $C_6=1$) of the general equation of a conic can be calculated solving the overdetermined system

$$ZH = I \tag{2}$$

where: $H = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 \end{bmatrix}$, $Z = \begin{bmatrix} x_{s,i}^2 & x_{s,i} z_{s,i} & z_{s,i}^2 & x_{s,i} & z_{s,i} \end{bmatrix}_{i=1,2\ldots6}^T$

Being (x_s,z_s) computer coordinates and I the 1x6 unit matrix. As was previously mentioned, *swing* angle corresponds to the angle (ψ) between the major axis of the ellipse and the vertical reference axis Z_s. *Swing* angle ψ can be easily determined from the estimated parameters , C_1, C_2 and C_3 from equation:

$$\psi = \frac{1}{2} a\,tan \frac{C_2}{C_1 - C_3}, C_1 \neq C_3; \quad \psi = 45^o, C_1 = C_3 \tag{3}$$

The camera-pattern distance can be calculated through the focal and the major axis, (See Figure 2), where a is the major axis of the ellipse the image plane of the camera, f is the focal of the camera and R is the distance from P_9 to whatever external dot. This expression proves that the camera-pattern distance does not depend on the other angular parameters (Figure 2 right).

$$D = \frac{f}{a} R \tag{4}$$

Tilt angle is obtained from the eccentricity of the ellipse following the last equation. Details about obtaining of this parameter can be found in [16].

$$sin\,\phi = e \tag{5}$$

e being the ellipse eccentricity. Therefore, tilt angle is obtained from the eccentricity of the ellipse fitted to the external points of the pattern. Values of ϕ are in the interval [0, 90º]. When $e=0$, $\phi=0$ and the points are fitted to a circle whereas when $e=1$, $\phi=90^o$ and the ellipse is converted into a segment.

Finally, *pan* parameter is obtained through the position of P_1 in the ellipse coordinate system. Let $(x_{e1}, y_{e1}, z_{e1}, 1)_{S_e}$ and $P_1 = (0,0,-R,1)_{S_0}$ be the image coordinates and the pattern coordinates of P_1. Using the transformation M we obtain

$$x_{e1} = -R\cos\theta \tag{6}$$

$$z_{e1} = -R\cos\phi\sin\theta \tag{7}$$

and taking into account that $\dfrac{x_e}{z_e} = \dfrac{x_s}{z_s}$ we finally obtain:

$$\tan\theta = \frac{z_{s1}}{x_{s1}\cos\phi} \tag{8}$$

Equation (3) is maintained. Although dot P_1 was one of the missing dots in the image, equation (8) is maintained for the dual dot P_5 and parameter θ can be obtained. Even if P_1 and P_5 are occluded, we can use both $P_3 = (R,0,0,1)_{S_0}$ and $P_7 = (-R,0,0,1)_{S_0}$ finding $tg\theta = -\cos\phi\dfrac{x_{s1}}{z_{s1}}$.

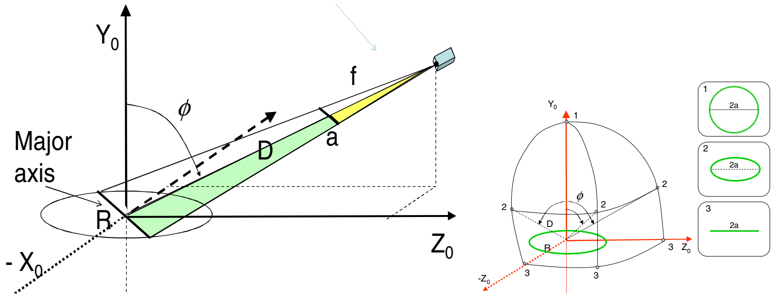


Fig. 2. Analysis of a circle projected in the image and invariance of the major axis length in a sphere around the pattern

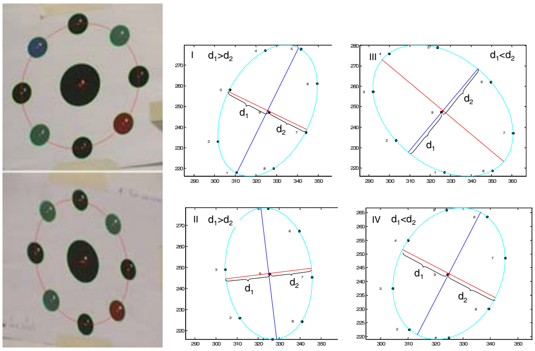


Fig. 3. Left) Images of the landmark from first and second quadrant and ellipses fitted to the dots (in red). Right) Aspect of the fitted ellipse depending on the quadrant in which the camera captures the image.

Table 1. Correction of parameter θ

	d1>d2	*d1<d2*
ψ>0	I	IV
	θ	*θ+π*
ψ<0	II	III
	θ+π	*θ*

When $\psi \neq 0$, non-rotated coordinates x'_s and z'_s must be substituted in the last equation. Equation (8) yields indeterminate values of θ. This problem can be solved by knowing the quadrant (in the system S_0), where the camera is placed. This quadrant is established through the sign of ψ and the position of the point P_9 in the minor axis. It can be proved that, due to projective reasons, P_9 is displaced with respect to the theoretical ellipse center. As a consequence of this, distances d_1 and d_2 from P_9 to the ellipse, in the minor axis direction, are different. Therefore, we can infer whether the pattern is viewed from the left (case $d_1 > d_2$) or from the right (case $d_1 < d_2$). Figure 3 illustrates the displacement of P_9. Table 1 shows the quadrant as well as the applied correction of parameter θ.

Level 2. This happens when more than two outer dots are occluded but the internal dot P_9 is in the image. In this case, equation (2) does not converge and a new strategy must be implemented. This frequently occurs for short user-pattern distances where the view angle of the camera is reduced and a small head movements made by the user can generate loss of the dots in the image. Since any visible circular dot can be viewed as an ellipse, we adapt the pose strategy presented in level 1 for dot P_9.

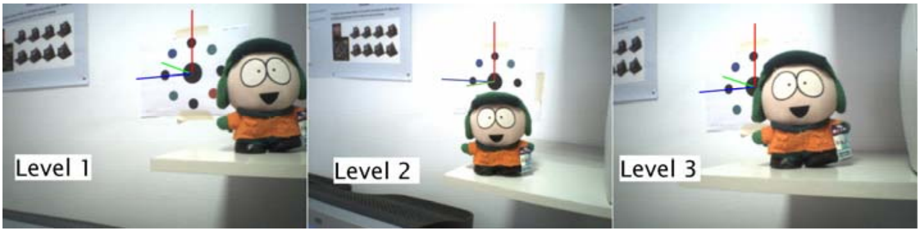


Fig. 4. Examples of several occlusion levels. As we can see, the landmark reference system retroprojected on the image according to the calculated camera positioning.

Level 3. More than two outer dots are occluded and the internal dot P_9 is also missing. This is the highest occlusion level and occurs when the user is near the pattern. After having identified several dots in the image and calculated the pose parameters for each one, we take a weighted mean as the best pose approximation where the weight depends on how close the dot is to the center of the image. Note that in this case, an error is introduced because the pose is calculated in a coordinate system which is translated with respect to S_0. Figure 4 shows examples of different occlusion levels.

4 Experimental Results

In order to prove the applicability of our method under real conditions, we have tested the pose algorithm imposing soft and severe occlusion. The approach was implemented in an autonomous augmented reality system which consists of a Trivisio AR-vision-3D HMD binocular head-mounted display with a color camera and a Quantum3D Thermite portable computer.

A user wearing the portable AR system searches for the pattern when he wants to know its current position in the world coordinate system. Then, the user can see, through one of the two displays of the HMD, the image of the camera and its current position in real-time. The other display is used for superimposing virtual information on the real scene.

The followings phases are repetitively executed on board: I) find/track the pattern in the image, II) segmentation of dots, III) calculate pose parameters. Depending on the occlusion level, we have parameters ψ, ϕ, θ, D' (levels 1 and 2) or parameters for each outer dot ψ_i, ϕ_i, θ_i, D_i' (level 3), IV) obtain a unique pose solution.

The system works with 640x480 images and spends 45 ms to take one frame, process the image and calculate the pose. Thus, the performance average rate is 23 frames/second. Two different environments - indoors and outdoors - have been tested while imposing occasional occlusions. The pattern was occluded by obstacles or people walking in front of the user's viewpoint. Several occlusion circumstances also occurred due to the proximity of the pattern or the user's rapid head movements.

Some information about the performance of the method is included below. Tables 2 and 3 summarizes statistical results of the errors obtained for each estimated user's coordinate and user-pattern distance for both no occlusion and occlusion cases. Absolute and relative errors are presented in two sub-tables. For each case, average, standard deviation, greatest and smallest errors are presented as well. Promising results has been obtained in both cases. Note that position error average was below 5cm for non-occluded case and below 7cm in case of partial occlusions. These results are acceptable enough in the framework we are carrying out in where the user observes the overlaps virtual models from distances always highest than one meter.

We have also designed similar patterns with higher dimension and with a color code which can easily identify each pattern in an extensive environment but this experimental report concerns non-colored patterns and performance under occlusion.

Table 2. Experimental results without occlusion

Abs. Errors (cm)	e(X)	e(Y)	e(Z)	e(D')
Average	2,4	2,33	4,76	2,60
Std.Dev	3,1	1,71	4,51	1,96
Greatest	10,1	6,5	13,6	5,6
Smallest	0,2	0,3	0,1	0,3
R. Errors (%)	e(X)	e(Y)	e(Z)	e(D')
Average	0,60	0,64	6,53	1,52
Std.Dev	0,51	0,45	6,49	1,56
Greatest	1,76	1,42	20,80	5,69
Smallest	0,03	0,06	0,17	0,11

Table 3. Experimental results with occlusion

Abs. Errors (cm)	e(X)	e(Y)	e(Z)	e(D')
Average	6,3	5,9	7,0	5,7
Std.Dev	4,0	3,0	3,7	2,1
Greatest	12,2	10,2	14,1	7,0
Smallest	1,1	0,4	2,0	0,91
R. Errors (%)	e(X)	e(Y)	e(Z)	e(D')
Average	1,31	1.67	4,43	1,98
Std.Dev	1,23	1,66	3,14	1,03
Greatest	3,05	5,02	15,23	3,77
Smallest	0,92	1,02	0,34	0,76

5 Conclusions

The method presented in this paper solves the location problem using a single camera on board an AR reality system. Until now, the majority of based-on-vision pose solutions concern mobile robots applications where the camera is onboard the robot having slow and controlled movements. For augmented reality applications, like ours, the camera is carried over a human head which involves quick and unexpected camera movements. In this sense, we propose a pose method in an unusual environment.

The pose is calculated after analyzing the projected image of an artificial landmark consisting of nine dots. Due to the simplicity of the pattern and the low computational cost in the image processing phase, the system is capable of working under on-line requirements in AR-based navigation applications. Furthermore, opposite to most of the landmark-based positioning solutions, the system works in severe occlusion circumstances and for a wide distance range which makes it more robust than other solutions.

Our approach is being used for AR systems in autonomous navigation for humans yielding excellent results. Experimentation, advantages and restriction of this technique have been illustrated in the paper.

References

1. Cobzas, D., Jagersand, M., Sturm, P.: 3D SSD tracking with estimated 3D planes. Journal of Image and Vision Computing 27, 69–79 (2009)
2. Duan, F., Wu, F., Hu, Z.: Pose determination and plane measurement using a trapezium. Pattern Recognition Letters 29(3), 223–231 (2008)
3. Feng, W., Liu, Y., Cao, Z.: Omnidirectional Vision Tracking and Positioning for Vehicles. In: ICNC 2008. Fourth International Conference on Natural Computation, vol. 6, pp. 183–187 (2008)
4. Fiala, M.: Linear Markers for Robots Navigation with Panoramic Vision. In: First Canadian Conf. Computer and Robot Vision, 2004. Proceedings, pp. 145–154 (2004)
5. Jang, G., et al.: Metric Localization Using a Single Artificial Landmark for Indoor Mobile Robots. In: International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), pp. 2857–2862 (2005)
6. Josephson, K., et al.: Image-Based Localization Using Hybrid Feature Correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
7. Neumann, U., et al.: Augmented Reality Tracking in Natural Environments. In: International Symposium on Mixed Reality, 1999. ISMR 1999, pp. 101–130 (1999)
8. Xu, K., Chia, K.W., Cheok, A.D.: Real-time camera tracking for marker-less and unprepared augmented reality environments. Image and Vision Computing 26(5), 673–689 (2008)
9. Se, S., Lowe, D., Little, J.: Mobile Robot Localization and Mapping with Uncertainly using Scale-Invariant Visual Landmarks. The International Journal of Robotics Research 21(8), 735–757 (2002)
10. Vachetti, L., Lepetit, V., Fua, P.: Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. In: Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004. ISMAR 2004, pp. 48–56 (2004)
11. Vachetti, L., Lepetit, V., Fua, P.: Stable Real-Time 3D Tracking using Online and Offline Information. IEEE Transactions on PAMI 26(10), 1385–1391 (2004)
12. Briggs, A.J., et al.: Mobile Robot Navigation Using Self-Similar Landmarks. In: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA 2000, vol. 2, pp. 1428–1434 (2000)
13. Kato, H., et al.: Virtual Object Manipulation on a Table-Top AR Environment. In: Proceedings of IEEE and ACM International Symposium on Augmented Reality, 2000 (ISAR 2000), pp. 111–119 (2000)
14. Koller, D., et al.: Real-time Vision-Based Camera Tracking for Augmented Reality Applications. In: ACM Symp. on Virtual Reality Software and Technology, pp. 87–94 (1997)
15. Hager, G.D., Belhumeur, P.N.: Efficient Region Tracking with Parametric Models of Geometry and Illumination. IEEE PAMI 20(10), 1025–1039 (1998)
16. Adan, A., Martín, A., Chacón, R., Dominguez, V.: Monocular Model-Based 3D Location for Autonomous Robots. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 594–604. Springer, Heidelberg (2008)

A Binarization Method for a Scenery Image with the Fractal Dimension

Hiromi Yoshida* and Naoki Tanaka

Kobe University
Graduate School of Maritime Science
070w706w@stu.kobe-u.ac.jp,
ntanaka@maritime.kobe-u.ac.jp

Abstract. We propose a new binarization method suited for character extraction from a sign board in a scenery image. The binarization is thought to be a significant step in character extraction in order to get high quality result. Character region of sigh board, however, has many variation and colors. In addition to it, if there exists high frequency texture region like a mountain or trees in the background, it can be a cause of difficulty to binarize an image. At the high frequency region, the binarized result is sensitive to the threshold change. On the other hand, a character region of sign board consists of solid area, that is, includes few high frequency regions, and has relatively high contrast. So the binarized result of character region is stabile against an interval of the threshold value. Focusing attention on this point, we propose a new method which obtains a threshold value based on the fractal dimension to evaluate both region's density and stability to threshold change. Through the proposed method, we can get a fine quality binarized images, where the characters can be extracted correctly.

Keywords: Binarization Fractal dimension Blanket method.

1 Introduction

Binarization of gray level image is a significant step in region extraction and a number of binarization methods have been proposed. Trier[1] evaluated 15 binarization methods as promising by the procedure called goal-directed evaluation, and showed that Niblack's method[2] has the best performance as a local adaptive method and Otsu method[3] is the best in global methods. In these 15 methods, threshold value is selected based on the local or global statistical information of the gray level such as gray level histogram. Scenery image which contains sign board consists of many regions such as high or low texture, solid area, and have a contrast perturbation. In an image which has much color variation, the binarization methods which use gray level information only can generate poor results. So we introduce evaluating method of a binarized image using fractal dimension. The FD has relatively large peaks for texture regions and a stable interval for a character string region, i.e. sign board, respectively.

[*] Corresponding author.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 29–36, 2009.

By detecting the stable interval of a FD, we can obtain a threshold value which is the best value to obtain a fine binarized character string. That the fractal dimension can evaluate the fineness of a binarized result is reported by Yoshida[4]. The method detects cracks from considerably noisy road surface images.

The proposed method corresponds to a preprocessing step of character extraction method in a scenery image which contains a sign board. The proposed new binarization method uses a fractal dimension for evaluating binarized image to find a threshold value. By using our method, we can evaluate the density of regions and stability to threshold change, that is, connected components which are close to each other and consist of sold areas in input images. First, about fractal dimension is described. Then, we show algorithm of our technique and its experimental result.

2 Associated Technique

Fractal dimension was proposed as a method of texture analysis named "Blanket method" by SHUMEL[5] in 1984. The range of the dimension is from 2 to 3 and is obtained from expression1,2,3 and 4.

$$U_\epsilon = \max\{U_{\epsilon-1}(i,j)+1, \max_{|(m,n)-(i,j)|\leq 1} U_{\epsilon-1}(m,n)\} \tag{1}$$

$$b_\epsilon = \min\{b_{\epsilon-1}(i,j)-1, \min_{|(m,n)-(i,j)|\leq 1} b_{\epsilon-1}(m,n)\} \tag{2}$$

$$A(\epsilon) = \frac{\sum_{i,j}(U_\epsilon(i,j) - b_\epsilon(i,j))}{2\epsilon} \tag{3}$$

$$A(\epsilon) = F\epsilon^{2-D} \tag{4}$$

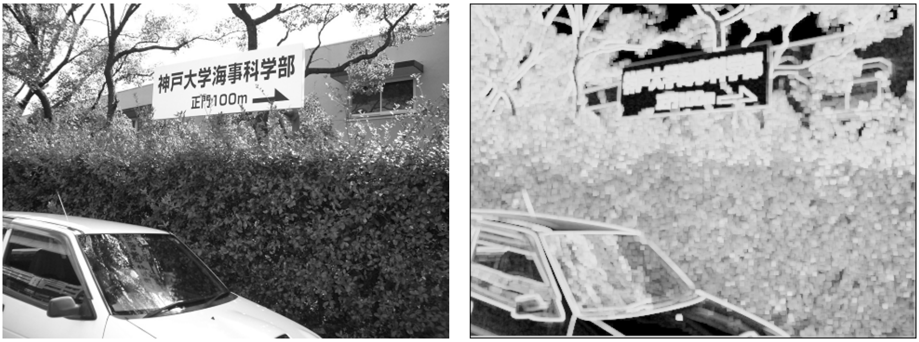
where ϵ is the number of blanket. Fractal dimension is calculated globally and also locally with a window. Noviant[6] proposed optimal range of the fractal dimension and adapts it locally to image and get a local fractal dimension (LFD) image. LFD image has a feature that brightness of region is proportional with frequency of texture region. Example of LFD images are shown at Fig.1. Window size for LFD is 3x3 and the blanket number is 44.

3 Binarization Algorithm

Proposed algorithm has 5 step procedures as follows.

- ***Step*1**: Binarize the input image $\boldsymbol{I(x)}$ with every threshold values from **0** to **255**, and obtained the 256 binarized images $\boldsymbol{I_{b_i}(x)}$ $\boldsymbol{(i = 0, 1, ..., 255)}$ respectively.

- ***Step*2**: The $\boldsymbol{FD(i)}$ values can be calculated on $\boldsymbol{I_{b_i}(x)}$ images by the Blanket method.



(a) Input image

(b) LFD image

Fig. 1. Example of LFD image

- ***Step*3**: We treat the $\boldsymbol{FD(i)}$ to be a function respect to $\boldsymbol{i}$. Then, smooth the $\boldsymbol{FD(i)}$ to remove noise of the function, and find the stable interval of $\boldsymbol{FD(i)}$ by taking the first derivation.
- ***Step*4**: In a stable interval, detect the minimum $\boldsymbol{FD(i)}$(which is not smoothed) as the threshold value $\boldsymbol{\theta}$.
- ***Step*5**: Final binarization Image $\boldsymbol{B(x)}$ can be obtained using the threshold value $\boldsymbol{\theta}$.

In the step3, we define the stable interval as follows: differentiate the smoothed $\boldsymbol{FD(i)}$ with respect to $\boldsymbol{i}$, and count the number of $\boldsymbol{\frac{\delta}{\delta i}FD(i)}$ which is equal $\mathbf{0}$ or nearly equal to $\mathbf{0}$ until it turns larger than $\mathbf{0}$. $\boldsymbol{\frac{\delta}{\delta i}FD(i)}$ is calculated from expression 5.

$$\boldsymbol{\frac{\delta}{\delta i}FD(i) = FD(i+1) - FD(i)} \tag{5}$$

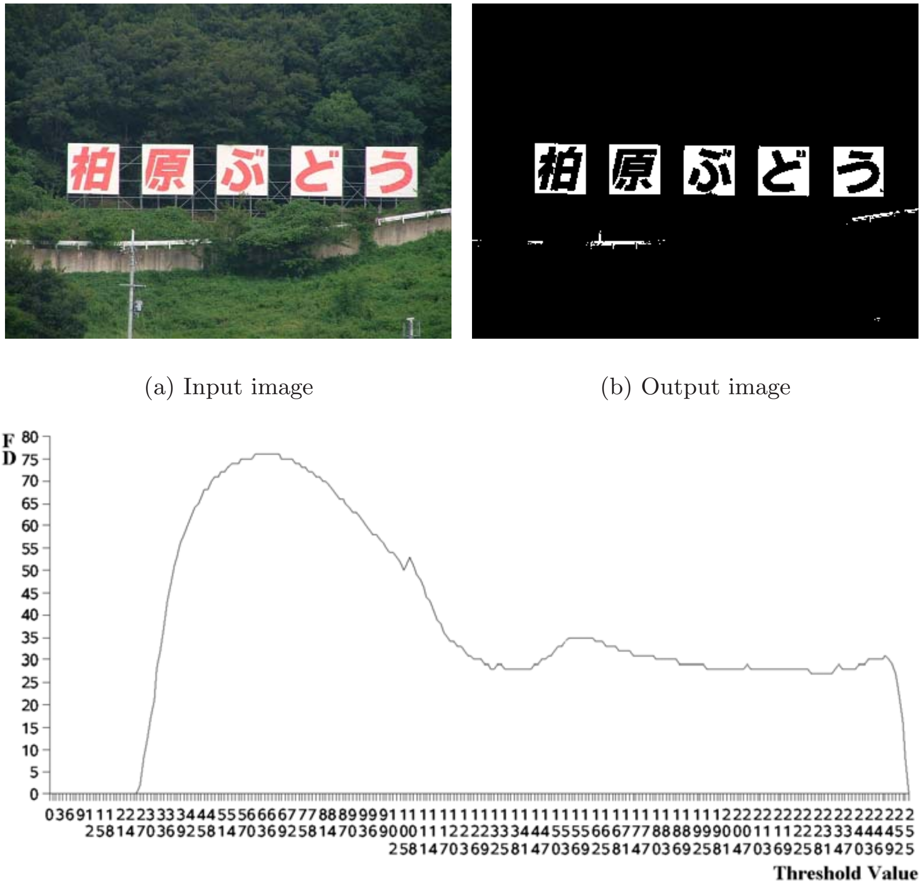
The longest interval where $\boldsymbol{\frac{\delta}{\delta i}FD(i)}$ is flat defined as the stable interval. Example of graph of $\boldsymbol{FD(i)}$ is shown at Fig.2.

4 Experiment

4.1 Experimental Parameter

Table 1 shows the parameters used in the experiment.

Noviant showed that appropriate range of ϵ is from 34 to 53. So we selected 44 as median of the range. Since ϵ in $\boldsymbol{FD(i)}$ relates to evaluating the density of a regions, the binarized results are gradually changing. Fig.3 showed how changing the binarized results. In the implementation , we use a ”double-precision floating-point data type” for $\boldsymbol{FD(i)}$ to decide a threshold value $\boldsymbol{\theta}$ precisely.



(a) Input image

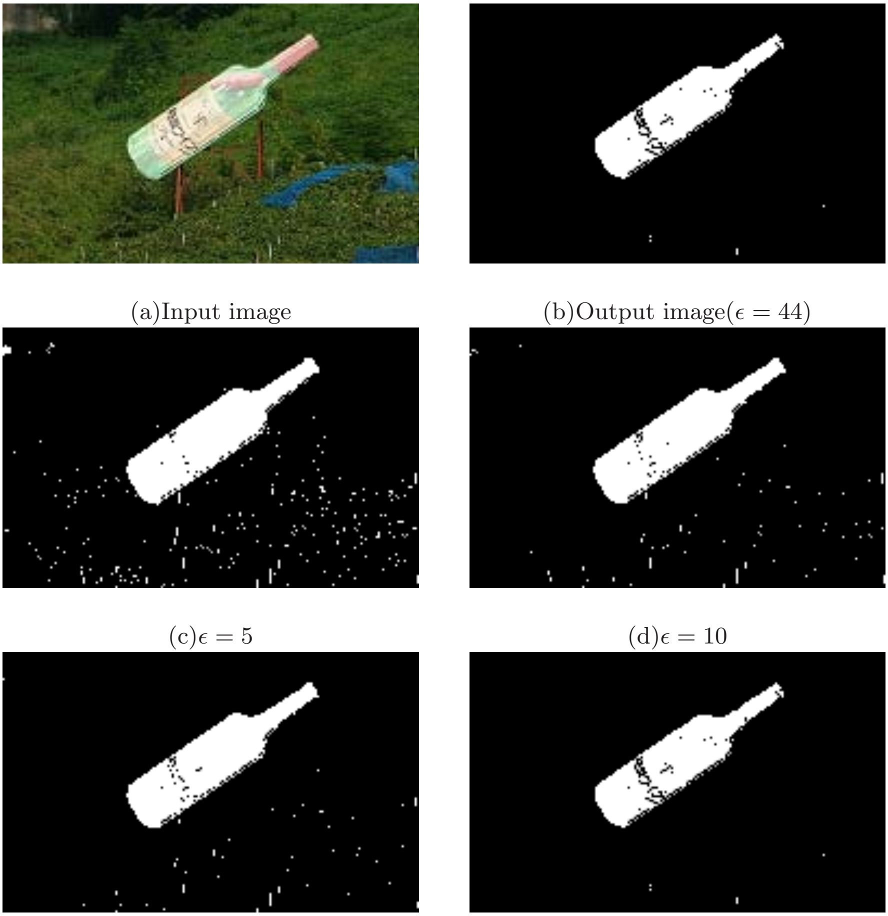
(b) Output image

(C) FD graph

Fig. 2. Graph of $\boldsymbol{FD(i)}$

Table 1. Experimental parameter

	smoothed $\boldsymbol{FD(i)}$	$\boldsymbol{FD(i)}$
ϵ	44	44
Range of quantization	100	100.0000



(a)Input image

(b)Output image($\epsilon = 44$)

(c)$\epsilon = 5$

(d)$\epsilon = 10$

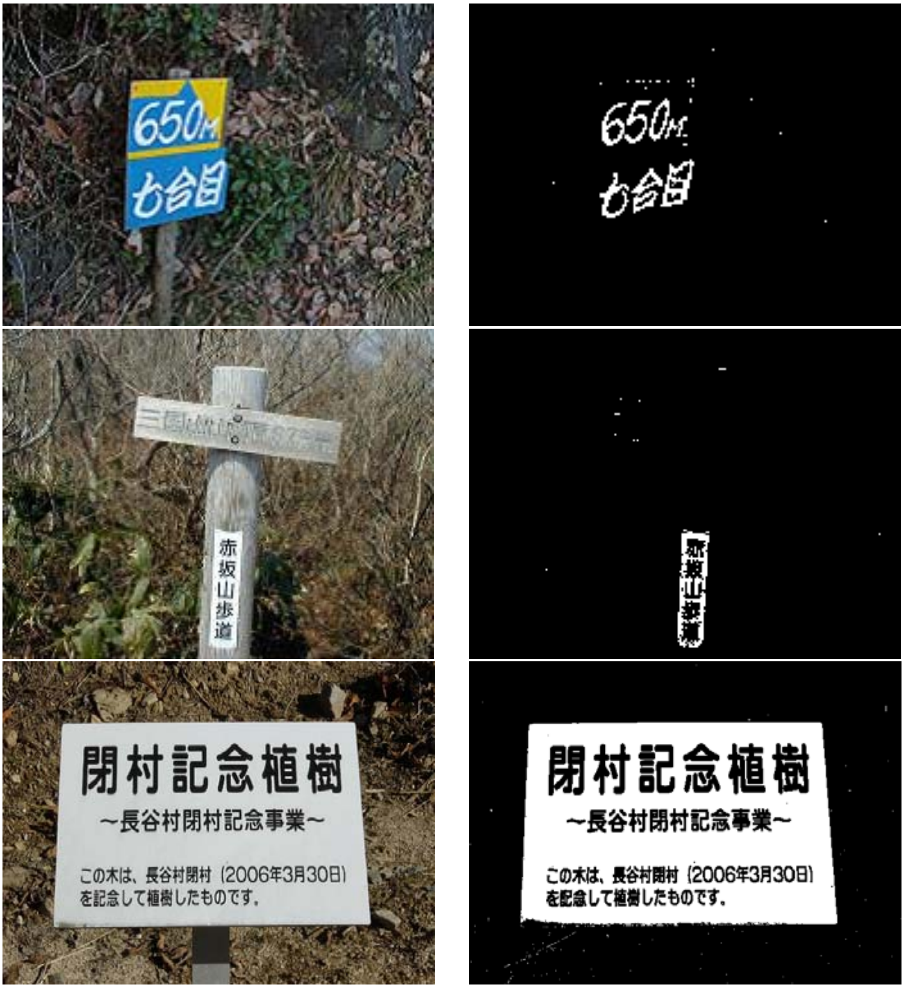
(e)$\epsilon = 25$

(f)$\epsilon = 45$

Fig. 3. The examples of difference result due to change of blanket number

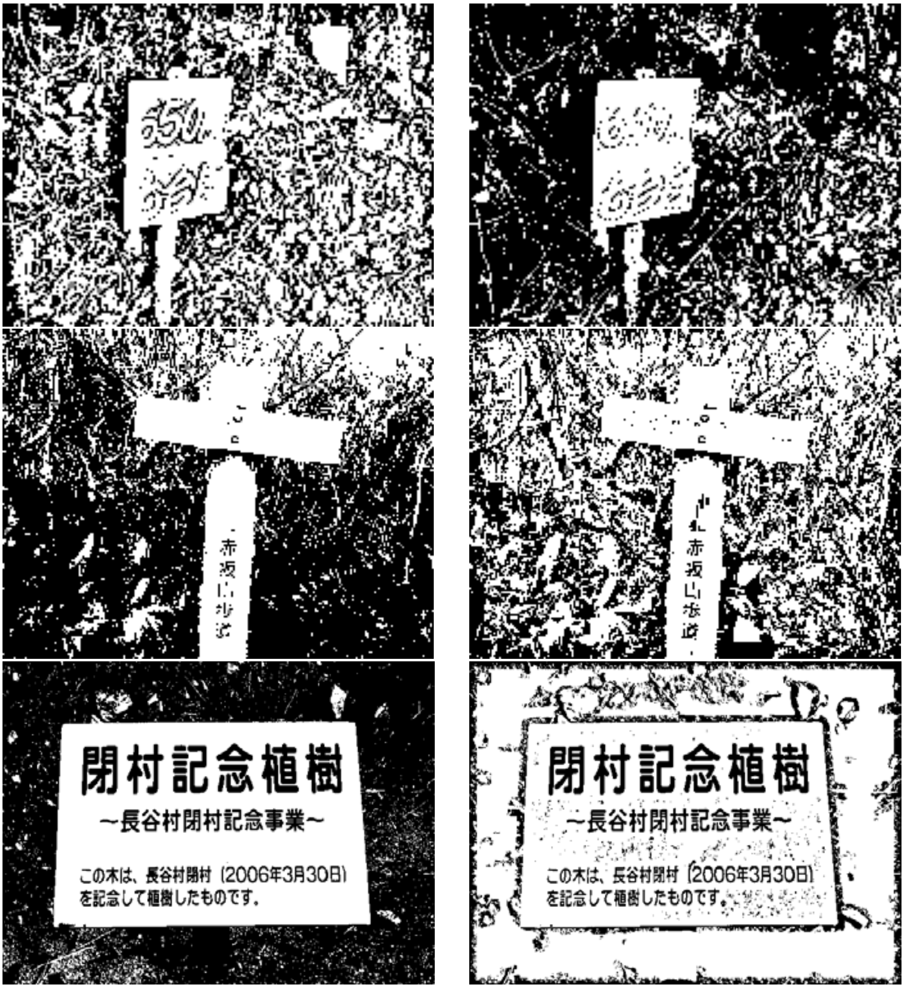
4.2 Experimental Result

Binarized images by proposed algorithm are shown at Fig.4. And processed results by Otsu's method and Niblack's method are also shown for comparison at Fig.5. We select these two methods because they have the best performance a promising binarization method using gray level information. The former is a local method; the latter is a global one.


650M
七合目
赤坂山歩道
閉村記念植樹
〜長谷村閉村記念事業〜
この木は、長谷村閉村（2006年3月30日）
を記念して植樹したものです。

(a)Input image (b)Output image($\epsilon = 44$)

Fig. 4. The examples of binarized images by proposed method


赤坂山歩道
閉村記念植樹
～長谷村閉村記念事業～
この木は、長谷村閉村（2006年3月30日）
を記念して植樹したものです。

(a)Otsu's method

(b)Niblack's method

Fig. 5. The examples of binarized images by Otsu's method and Niblack's method

5 Conclusion

Result of experiment shows that the proposed method can be a promising binarization step of character detection method from sign board in a scenery image. Some noises are still remaining in binarized image. This is because our proposed method belongs to global techniques, that is, only one threshold value is applied for whole image. So we will develope a local adaptive binarization method based on this technique, and hope to get still improve the results.

References

1. Trier, O.D., Jain, A.K.: Goal-Directed Evaluation of Binarization Methods. IEEE Trans. Pattern Analysis and Machine Inteligence 7(12), 1191–1201 (1995)
2. Niblack, W.: An Introduction to Digital Image Processing, pp. 115–116. Prentice Hall, Englewood Cliffs (1986)
3. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans, Systems, Man, and Cybernetics 9(1), 62–66 (1979)
4. Yoshida, H., Tanaka, N.: A Binarization method for Crack Detection in a Road Surface Image with the Fractal Dimension. In: Proceedings of MVA 2009 IAPR Conference on Machine Vision Applications, pp. 70–73 (2009)
5. Peleg, S., Naor, J., Hartley, R., Avnir, D.: Multiple resolution texture analysis and classification. IEEE Trans. Pattern Analysis and Machine Inteligence 6(4), 518–523 (1984)
6. Noviant, S., Suzuki, Y., Maeda, J.: Optimumestimation of local fractal dimension based on the blanket method. IPSJ Journal 43(3), 825–828 (2002)
7. Wang, Y., Tanaka, N.: Text String Extraction from Scene Image Based on Edge Feature and Morphology. In: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems, pp. 323–328 (2008)
8. Yoshida, H., Tanaka, N.: A Study on Signboard Image Identification with SIFT Features. In: Handout of 8th IAPR International Workshop on Document Analysis Systems, pp. 10–14 (2008)

Selective Change-Driven Image Processing: A Speeding-Up Strategy

Jose A. Boluda[1], Francisco Vegara[2], Fernando Pardo[1], and Pedro Zuccarello[1]

[1] Departament d'Informàtica, Universitat de València. Avda Vicente Andrés Estellés, s/n. 46100-Burjassot, Spain
[2] Institut de Robòtica, Universitat de València. Polígono de la Coma, s/n. Aptdo. 2085, 46890-València, Spain
{Jose.A.Boluda,Francisco.Vegara,Fernando.Pardo,Pedro.Zuccarello}@uv.es
http://tapec.uv.es

Abstract. Biologically inspired schemes are a source for the improvement of visual systems. Real-time implementation of image processing algorithms is constrained by the large amount of data to be processed. Full image processing is many times unnecessary since there are many pixels that suffer a small change or not suffer any change at all. A strategy based on delivering and processing pixels, instead of processing the complete frame, is presented. The pixels that have suffered higher changes in each frame, ordered by the absolute value of its change, are read-out and processed. Two examples are shown: a morphological motion detection algorithm and the Horn and Schunck optical flow algorithm. Results show that the implementation of this strategy achieves execution time speed-up while keeping results comparable to original approaches.

1 Introduction

Full image processing is usually the classical approach for general image sequence processing, where each image is a snapshot taken at regular intervals. The normal procedure implies the application of the processing algorithm for each image in the sequence. Biological systems work in a different way: each sensor cell sends its illumination information independently. It is possible to reduce processing time by taking into account that images usually change little from frame to frame, especially if the acquisition time is short. This is particularly true in motion detection algorithms with static cameras.

A biologically inspired camera would send pixel information asynchronously when changes are produced, rather than full acquired images. Following this ideas, it is also possible to implement a change-driven data-flow policy in the algorithm execution, processing only those pixels that have changed. Paying attention only to those pixels that change is not new and this principle has been employed to design some image sensors with on-plane compression [1]. These image sensors only deliver the pixels that change, decreasing the amount of data coming from the camera. This strategy will decrease the total amount of data to be processed; consequently also it will decrease the number of instructions and thus the computer execution time.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 37–44, 2009.

A biologically motivated global strategy for speeding-up motion detection algorithms is presented. The system includes a change-driven camera that delivers pixels change instead of synchronous full images, and a data-flow algorithm adaptation for the image processing algorithm.

2 Change-Driven Camera and Processing

2.1 Change-Driven Camera

Biological visual systems has been already partially emulated taking into account its asynchronous nature [2]. Each pixel works independently in this visual sensor and the available output bandwidth is allocated according to pixel output demand. In this kind of sensors the change event signaling depends on a contrast sensitivity threshold, which is also found in biological vision systems. A pixel change greater than this threshold is considered as a change, consequently this pixel is read out and processed. This threshold has already been successfully employed to accelerate differential movement algorithms [3].

A Selective Change-Driven (SCD) camera with pixel delivering for high-speed motion estimation is under construction [4]. In this camera every pixel has an analogue memory with the last read-out value. The absolute difference between the current and the stored value is compared for all pixels in the sensor; the pixel that differs most is selected and its illumination level and address are read out for processing. With this strategy, every pixel that has changed will be sent sooner or later, and thus processed in a data-flow manner, ordered by its illumination change.

2.2 Data-Flow Processing

A generic image processing algorithm can be programmed as an instruction sequence within a classical control flow computing model. The data-flow model works in a totally different way: instructions are fired when the data needed for these instructions are available [5]. One of the main advantages of this model is the reduction of the instructions to be executed when little changes are produced in input data.

Motion detection algorithms (and as a particular case, differential algorithms) greatly benefits from the approach of firing instructions of an algorithm only when data changes. Often only few pixels change from frame to frame and usually there is no need to execute any instruction for unchanged pixels. The classical approach performs the same calculation for all the pixels in an image for every image in a sequence, even if the pixel did not change at all. It is possible to save many calculations if only those pixels that have changed are delivered by the SCD camera and fire the related instructions.

Any image processing algorithm would need to be rebuilt in order to be implemented following the SCD strategy. Extra storage to keep track of the intermediate results of preceding computing stages is needed by this methodology.

3 Motion Detection Algorithms

The change-driven delivering and processing has been tested in several motion detection algorithms that have been rebuilt in a data-flow manner. Two examples are included in this paper.

3.1 Traffic Detection Algorithm

This motion detection algorithm has already been utilized serving as an example of how the use of a change sensitive threshold can accelerate differential motion detection algorithms [3].

A detailed description of the original sequential procedure can be seen at [6], where I_t is the input sequence and M_t is the estimated background value. The estimate background is increased by one at every frame when it is smaller than the sample or decreased by one when it is greater than the sample. The absolute difference between I_t and M_t is the first differential estimation $\varDelta_t$, that is used to compute the pixels motion activity measure, employed to decide whether the pixel is moving or static. V_t is used as the dimension of a temporal standard deviation. It is computed as a $\varSigma - \varDelta$ filter of the difference sequence. Finally in order to select pixels that have a significant variation rate over its temporal activity, the $\varSigma - \varDelta$ filter is applied $N = 4$ times. A simple common edges hybrid reconstruction is performed to enhance $\varDelta_t$ as shown in equation (1). The inputs are the original image I_t and the $\varSigma - \varDelta$ difference image $\varDelta_t$.

$$\varDelta'_t = HRec_{\alpha}^{\varDelta_t}(Min(\|\nabla(I_t)\|, \|\nabla(\varDelta_t)\|)) \tag{1}$$

The gradient modules of $\varDelta_t$ and I_t are computed by estimating the first Sobel gradient and then computing the Euclidean norm. $Min(\|\nabla(I_t)\|, \|\nabla(\varDelta_t)\|)$ acts as a logical conjunction, retaining the edges that belong both to $\varDelta_t$ and I_t.

The common edges within $\varDelta_t$ and with α as structuring element (a ball with radius=3) are reconstructed in order to recover the object in $\varDelta_t$. This is done by performing a geodesic reconstruction of the common edges (marker image) with $\varDelta_t$ as reference. Thus, after $\varDelta_t$ has been reconstructed, V_t and D_t are computed.

Change-Driven Data-Flow Algorithm. The original algorithm has been modified using the change-driven data-flow processing strategy. With this procedure as soon as the SCD camera delivers a pixel that has changed $\varDelta I(x, y)$ the related instructions are fired, and the intermediate images are updated. Fig. 1 shows the data-flow and the intermediate stored images.

An initial image is stored in the computer as the current image. Any absolute difference of the current image pixel with the stored image fires the intermediate images computation. Moreover, these updates must be done taking into account that the change of an input pixel may modify several output variables. For example, if a pixel is modified then its contribution to 6 pixels for the Sobel gradient image G_x and also for 6 pixels for image G_y must be updated. It may appear that a single pixel modification can produce too many operations, but

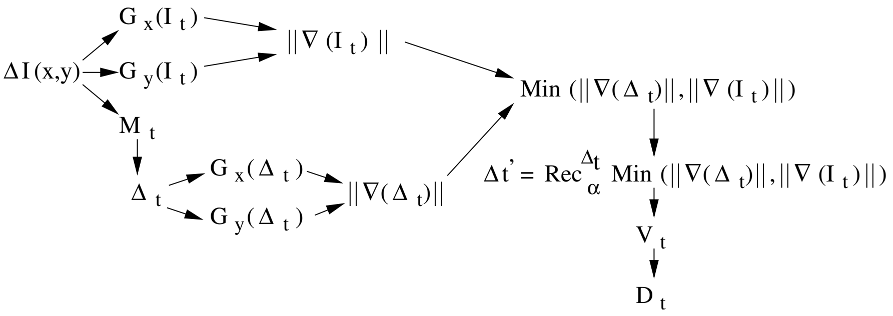


Fig. 1. Change-driven motion detection algorithm

theese are simple additions (and sometimes a multiplication by two) per pixel and they can be reutilized. In the original algorithm for each pixel the Sobel gradient images G_x and G_y are computed in any case; with six additions per pixel with the corresponding multiplications.

3.2 Horn and Schunk Optical Flow Computation

Optical flow is one of the main methods to estimate movement of objects and its calculation provides valuable information to artificial and biological systems. Unfortunately it is computationally intensive which constrains its use in real-time applications; despite of this, its high scientific interest motivates research on new strategies and hardware approaches to reduce its calculation time [8].

Differential techniques are applied under global restrictions in the Horn and Schunck method. Several global approximations are assumed as the conservation of intensity. Additionally, a method of global restriction that minimizes the squared magnitude of the gradient of the optical flow is introduced. A mask is used for Laplacian calculation of the mean values $(\bar{u}, \bar{v})$ of the optical flow components at any point (x, y), which are used in the equations that relate the optical flow vector from the image Laplacian and the spatial-temporal gradients:

$$u = \bar{u} - I_x \frac{I_x\bar{u} + I_y\bar{v} + I_t}{\lambda^2 + I_x^2 + I_y^2} \qquad v = \bar{v} - I_y \frac{I_x\bar{u} + I_y\bar{v} + I_t}{\lambda^2 + I_x^2 + I_y^2} \tag{2}$$

Final classical determination of the optical flow is done from these equations through an iterative full image processing over pairs of consecutive images.

Data-Flow version. The change-driven data-flow implementation of the Horn and Schunck algorithm uses the equations that treat only the data which are involved because of the variation of a given pixel, differently to the calculation of optical flow in the whole image. The procedure is as follows:

- Initial gradient and optical flow maps are computed for the whole image following the classical Horn and Shchunk method described before.

- Then, the changing pixels sent by the SCD sensor are processed in decreasing order of variation and for each received pixel the following operations are performed:
  - Recalculate the spatial and temporal gradients for those pixels of the image that are under the influence of the pixel that has changed.
  - Recalculate $(\bar{u}, \bar{v})$ for all pixel involved by the variation of pixel (i, j).

4 Experimental Results

Since the construction of the SCD camera is still in progress, the synchronous delivering of the pixels that have changed has been simulated by software. The comparison between the original algorithms and the change-based data-flow versions follows.

4.1 Traffic Detection Algorithm Results

The original traffic detection algorithm and the change-driven data-flow versions have been implemented. Both algorithms have been tested using several traffic sequences downloaded from the professor H. H. Nagel public ftp site: `http://i21www.ira.uka.de/image_sequences/` at the University of Karlsruhe. Fig. 2(a) shows a sequence frame of 740 × 560 pixels. Fig. 2(b) shows the original version with a full frame processing (414,400 pixels per frame). The change-driven data-flow algorithm results are shown in Fig. 2(c) with a mean of 80,000 pixels (roughly a 20% of image pixels). In this experiment, it has been simulated that the SCD camera has delivered out, ordered by the absolute magnitude of its change, a mean of 80,000 pixels.

Both result images shown at Fig. 2 are almost the same. There are no evident differences in terms of detected moving points between the original and the change-driven implementations. The executed time decreases significantly as Fig. 4 (left) shows (speed-up of 1.57). In this way, it must be appointed that the change-driven data-flow implementation gives similar results as the original, but with lower computational cost.

The execution time decreases because only changing pixels are processed. Moreover, if there are bandwidth limitations, the pixels with a bigger change

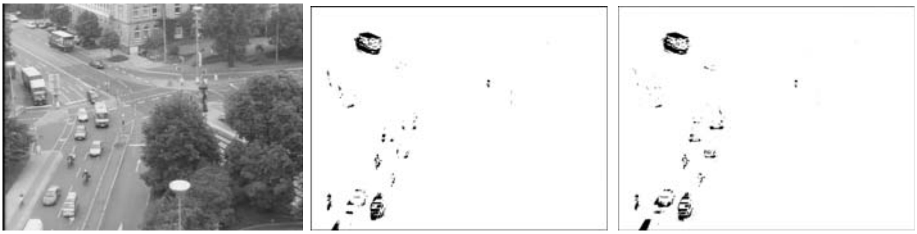


Fig. 2. (a) Original sequence, (b) Original algorithm results, (c) Change-Driven modified algorithm with 80,000 pixels (20%)

are processed first leaving as no processed the pixels with a lower change. The question in the application to this algorithm is whether there is a limit to decreasing the number of pixels and therefore to the algorithm speed-up versus the original full processing implementation. The answer is that if very few points are processed there is not a systematic background update, and moreover, moving points with a small gray value difference are not detected. This property produces two effects that limit the change-driven data-flow strategy. If there are more changes than pixels that the SCD camera can deliver due to bandwidth limitations, fewer correct moving points are detected and, otherwise, more false positives are detected. Values under a 10% of the image total number of pixels make this approach unfeasible and therefore further pepper noise filtering is required for D_t, disappearing the change-driven data-flow speed-up.

4.2 Change-Driven Optical Flow Results

The well known Rubik sequence has been downloaded from the ftp public site of the Department of Computer Science (`ftp://ftp.csd.uwo.ca/pub/vision/`) at the University of Ontario. Each frame has 61,440 pixels (256×240). Classical parameters have been used to calculate the optical flow: 10 iterations for each frame; only the flow vectors with modulus bigger or equal than 0.2 have been represented; the value of the Lagrange multiplier for the regularization term has been taken as $\lambda = 5$. Results are shown in the Fig. 3 for the classical algorithm implementation and for the change-driven implementation with 4,000 pixels, roughly a 7% of the image size.

Table 1 shows the mean angular deviation (in degrees) between the original Horn and Schunk algorithm and the change-driven implementation for different number of processed pixels. This angular deviation Ψ_E between the real velocity vector components (u_c, v_c) and the calculated velocity vector (u_e, v_e) has been computed through the optical flow error equation:

$$\Psi_E = \arccos\left(\frac{u_c u_e + v_c v_e + 1}{\sqrt{(u_c^2 + v_c^2 + 1)(u_e^2 + v_e^2 + 1)}}\right) \tag{3}$$

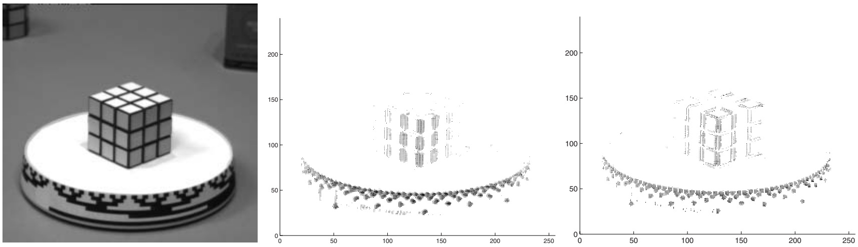


Fig. 3. (a) Original sequence, (b) Original optical flow results, (c) change-driven optical flow computed with 4000 pixels

Table 1. Mean error and standard deviation with different number of pixels

N^o of pixels	Mean error (^{o})	Standar deviation (^{o})
1000	18	8
2000	16	8
3000	15	8
4000	14	7
5000	13	7

The error shown at table 1 decreases as long as the number of pixels increases. In this algorithm, for a number of pixels under a 10%, result seems not very good but are not far from most optical flow algorithms with greater complexity. In this case, a lower number of pixels processed than in the traffic algorithm can give acceptable results. The Rubiks sequence has been taken in a controlled environment, giving less changing pixels.

Experimental measured speed-up of the change-driven optical flow algorithm referenced to the classical one is shown in Figure 4 (right). The optical flow is calculated for every received pixel as long as there is sufficient time until the next integration period. If not all received pixels can be processed, those with a bigger change in their luminance are processed first, since this is the way they will arrive from the sensor. This can be interpreted as an optical flow calculation at the pixel level instead of at the frame level. As expected, for a low number of processed pixels, there is a significant speed-up. If the number of pixels increases, the speed-up decreases. With 4,000 pixels there is still a speed-up of 1.2 with an error of 14° which can be useful for real-time applications.

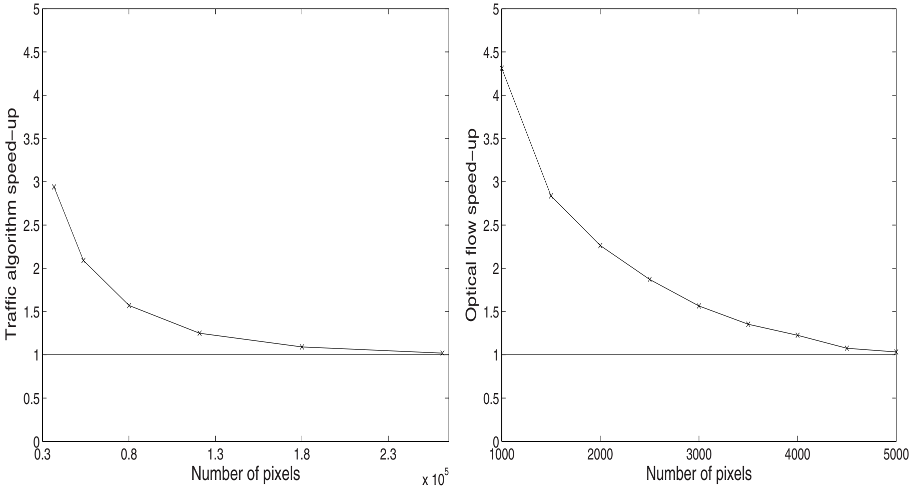


Fig. 4. Speed-up for the traffic detection (left) and the optical flow (right) algorithms

5 Conclusion

A biologically inspired strategy for speeding-up motion detection algorithms has been presented. The system includes an SCD camera that sends the pixel changes instead of sending sequentially full frames. Following these ideas, a generic image processing algorithm must be rebuild in a data-flow manner. The change-driven data-flow strategy is based on processing the pixels that have changed ordered by its absolute difference value.

The implementation of this methodology requires several algorithm adaptations and extra storage to keep track of the intermediate results. Two motion analysis algorithms have been chosen to test the change-driven data-flow policy: a morphological traffic detection algorithm and the Horn and Schunk optical flow algorithm. Change-driven data-flow algorithm implementations show a useful speed-up giving similar results of the original implementation.

Acknowledgments

This work has been supported by the projects TEC2006-08130/MIC and TEC2009-12980 of the Spanish *Ministerio de Ciencia e Innovación* and FEDER funds.

References

1. Özalevli, E., Higgins, C.M.: Reconfigurable Biologically Inspired Visual Motion Systems Using Modular Neuromorphic VLSI Chips. IEEE Transactions on Circuits and Systems 52(1), 79–92 (2005)
2. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128x128 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. IEEE Journal of Solid-State Circuits 43(2), 566–576 (2008)
3. Boluda, J.A., Pardo, F.: Speeding-up differential motion detection algorithms using a change-driven data-flow processing strategy. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 77–84. Springer, Heidelberg (2007)
4. Pardo, F., Boluda, J.A., Vegara, F., Zuccarello, P.: On the advantages of asynchronous pixel reading and processing for high-speed motion estimation. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 205–215. Springer, Heidelberg (2008)
5. Silc, J., Robic, B., Ungerer, T.: Processor architecture: from dataflow to superscalar and beyond. Springer, Heidelberg (1999)
6. Manzanera, A., Richefeu, J.C.: A new motion detection algorithm based on Σ-Δ background estimation. Pattern Recognition Letters 28, 320–328 (2007)
7. Teng, C.H., Lai, S.H., Chen, Y.S., Hsu, W.H.: Accurate optical flow computation under non-uniform brightness variations. Computer Vision and Image Understanding 97(3), 315–346 (2005)
8. Díaz, J., Ros, E., Pelayo, F., Ortigosa, E.M., Mota, S.: FPGA-based real-time optical-flow system. IEEE Transactions on Circuits and Systems for Video Technology 16(2), 274–279 (2006)

Coding Long Contour Shapes of Binary Objects

Hermilo Sánchez-Cruz and Mario A. Rodríguez-Díaz

Departmento de Sistemas Electrónicos. Centro de Ciencias Básicas
Universidad Autónoma de Aguascalientes. Av. Universidad 940
C.P. 20100. Aguascalientes, Ags. México
hsanchez@correo.uaa.mx

Abstract. This is an extension of the paper appeared in [15]. This time, we compare four methods: Arithmetic coding applied to 3OT chain code (Arith-3OT), Arithmetic coding applied to DFCCE (Arith-DFCCE), Huffman coding applied to DFCCE chain code (Huff-DFCCE), and, to measure the efficiency of the chain codes, we propose to compare the methods with JBIG, which constitutes an international standard. In the aim to look for a suitable and better representation of contour shapes, our probes suggest that a sound method to represent contour shapes is 3OT, because Arithmetic coding applied to it gives the best results regarding JBIG, independently of the perimeter of the contour shapes.

Keywords: Efficiency, Arithmetic coding, Chain code, Contour, Shapes, Binary objects.

1 Introduction

The shape representation of binary objects, is an active research in computer vision, pattern recognition and shape analysis. Binary objects can be seen as bi-level images, because they also are composed of two tones: black and white (B/W). Chain code techniques can be used to represent shape-of-objects in a right discretized fashion. It has been reported interesting applications using chain codes, for example: Mckee and Aggarwal [1] have used chain coding in the process of recognizing objects. Hoque et al. [2] proposed an approach to classify handwritten characters, based on a directional decomposition of the corresponding chain-code representation.

A chain code can be viewed as a connected sequence of straight-line segments with specified lengths and directions [3]. Chain codes can also be used to identify corners in shapes [4]. Salem et al. [5] discuss the capabilities of a chain code in recognizing objects. In [6], Sánchez-Cruz and Rodríguez-Dagnino proposed a code contour shape, called 3OT, and they found better compression properties than Freeman codes.

To compress binary objects, Liu and Zalik proposed the Differential Freeman Chain Code of Eight Directions (DFCCE) by using Huffman algorithm [7]. Liu et al. [8] introduced three new chain codes based on the VCC [9]. The main reason for the popularity of chain codes is their compression capabilities. There are two main categories for image compression algorithms, namely algorithms with loss of information such as

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 45–52, 2009.

MPEG, JPEG, etc., and lossless compression algorithms, such as JBIG, Huffman, Arithmetic and Lempel-Ziv (LZ). For instance, LZ algorithms are some of the most successful methods for text compression [10] and [11]. Aguinaga et al. [12], compared different entropy coding schemes applied to bi-level images by using run-length codes.

Another evidence to support that is better to codify contours shapes with 3OT codes instead than DFCCE, is by considering next analysis. In [8], three codes were introduced: EVCC, VVCC and C_VCC, in such a paper, C_VCC is found as the best compressor. In the spirit to look for better performance than C_VCC, Sanchez-Cruz, et al. [13], compared C_VCC with 3OT by doing some assignments, when changing appropriately the length of the 3OT chains. The procedure is easy: when contours are coded by 3OT, for every five consecutive 0's, a symbol "3" is introduced; for every five consecutive 1's the symbol "4" is introduced; and for every substring 0110 in the 3OT, a "5" symbol is introduced in a new alphabet: M_3OT = {0,1,2,3,4,5}, where each symbol has been obtained by doing the next assignments from 3OT chains:

$$\begin{gathered}0 \longrightarrow 0\\ 1 \longrightarrow 1\\ 2 \longrightarrow 2\\ 00000 \longrightarrow 3\\ 11111 \longrightarrow 4\\ 0110 \longrightarrow 5\end{gathered}$$

These new assignments permit us to improve the most recent code known as C_VCC. So, this fact constitutes another evidence that supports to utilize 3OT chain code to represent contour shapes.

An international committee has generated a standard image compressor for bi-level images called Joint Bi-level Image Experts Group (JBIG), which was primarily designed for compression without loss of information, [14]. JBIG has already been improved and the new standard is now called JBIG2 (see [16] and [17]). Sanchez-Cruz et al., [15] compared seven recent chain codes, including the 3OT and, also, JBIG; after they applied Huffman coding to the chains. Their experiments gave better results than the JBIG compressor. They found that the best codes to represent binary objects was DFCCE in comparing with JBIG, if considering a threshold, no more than about 13000 in Perimeter-8. We developed our research by using DFCCE (Differential Freeman Chain Code of Eight Directions) and 3OT (Three Orthogonal Change Directions), because they were the two best codes in [15], and were also compared with JBIG standard. 3OT is composed of three symbols, and we probe here that is suitable to be handled by Arithmetic coding, better than those composed with more symbols, including DFCCE, which has eight symbols. The contribution of our work is to find that there is not a limit in the contour shapes to be represented by 3OT code, whereas in [15] it was found that DFCCE was the best, however, an evident limit in contour perimeters were reported in such a paper.

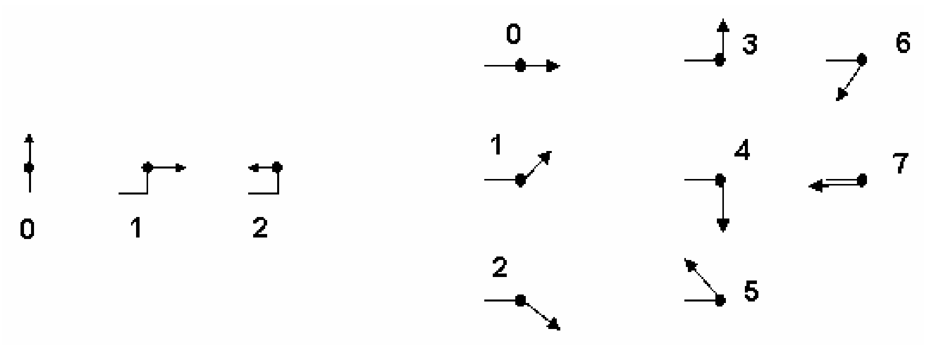
In this work, we utilized Arithmetic coding to 3OT and DFCCE chains; we found that this method has better performance than JBIG, and compress efficiently irregular objects in a 100%, of so large contours, even larger than the obtained until now, increasing the limit in perimeter found by [15].

We want to make it clear, we are not proposing replace JBIG by our method, but to use it as a standard to compare the efficiency of chain codes.

In Section 2, we explain the method proposed, whereas in Section 3 we give the results, and in Section 4 we give conclusions and further work.

2 Applying 3OT and DFCCE Chain Codes to Bi-Level Images

With the objective to compare the results with that found in [15], in this work, we calculate the perimeter-8 of a shape, that is given by squares as resolution cells and 8-connected neighborhoods. Fig. 1 shows the 3OT and DFCCE codes to represent the contours using resolution cells, as explained in [15].



(a) (b)

Fig. 1. The utilized codes: a) 3OT and b) DFCCE

Once the 3OT and DFCCE codes are obtained, we apply Arithmetic algorithm to the resulted chain. To compare with the proposed compression based on Huffman algorithm applied to DFCCE, we also computed such a method. Thus, we obtain an amount M_{CODE}, given in bytes. Also, we apply the JBIG compressor and obtain M_{JBIG}, in bytes too.

We encode and represent contour shapes for a variety of irregular sample objects, given in Fig. 2. The original size information is given in Table 1.

Table 1. Size of the bi-level images

Object	*Size*
Ant	235×250
Bat	551×267
Btrfy	667×822
Btrfly2	600×451
Bus	300×250
Camel	640×726

Table 1. (*Continued*)

Object	*Size*
Snail	640×633
Coco	302×87
Football	640×850
Dog	694×851
Horse	814×600
Lion	382×380
Plane	1801×1039
Map	648×648
Moto	960×738
Skull	1391×1333

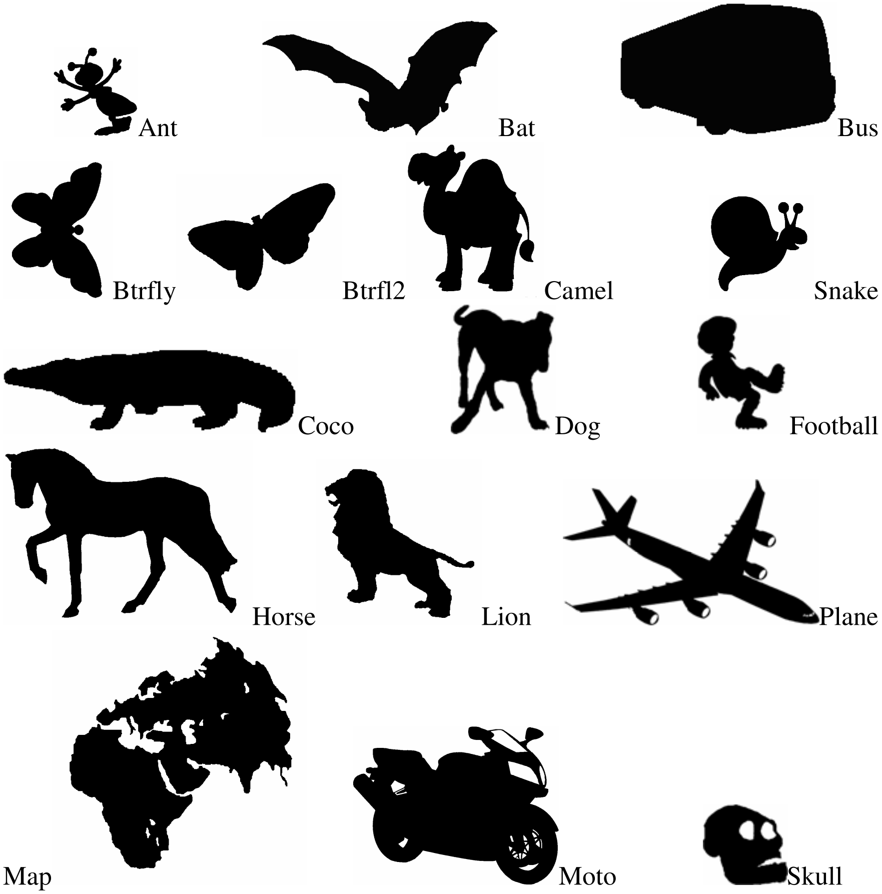


Fig. 2. Sample object shapes represented by chain codes. The actual scales appear in Table 1.

Of course, in the bi-level image, shape-of-objects are confined in a minimal rectangle of size $M \times N$. Notice that the smallest image corresponds to the Ant object, whereas Moto's shape almost fill a typical current screen of 14 inches (with a resolution of 1024×768 pixels, for example), and Plane and Skull do not fix into such a screen.

3 Results

To analyze our results, let us define the compression efficiency, regarding JBIG.

Definition. Let $Efficiency = 1 - M_{CODE} / M_{JBIG}$ be the compression efficiency of 3OT code with regard to the JBIG standard.

In Table 2 and 3 are reported the main results of this work. The values of perimeter-8 are given, also the storage memory due to 3OT, DFCCE and JBIG and the relative efficiency of 3OT with regard to JBIG. As can be seen Arithmetic to 3OT and to DFCCE improve compression levels, and are better than both: Huff-DFCCE and JBIG.

Table 2. Length chains, given in perimeter-8, of the coded contour shapes, the storage memory in bytes, and also, Efficiency regarding JBIG of the different codes

Object	*P-8*	JBIG	M_{DFCCE} *(Huffman)*	M_{DFCCE} *(Arith)*	M_{3OT} *(Arith)*	*Efficiency (Arith-3OT)*	*Efficiency (Aritc-DFCCE)*	*Efficiency (Huffman-DFCCE)*
Ant	1484	398	336	311	**309**	0.22	0.22	0.16
Bat	1444	392	323	297	**284**	0.28	0.24	0.18
Btrfly	2682	694	608	532	**507**	0.27	0.23	0.12
Btrfl2	1473	439	328	306	**283**	0.36	0.30	0.25
Bus	653	205	133	129	**115**	0.44	0.37	0.35
Camel	3446	746	715	662	**659**	0.12	0.11	0.04
Snail	2557	658	548	512	**502**	0.24	0.22	0.17
Coco	773	230	159	154	**145**	0.37	0.33	0.31
Football	3482	817	728	**674**	702	0.14	0.18	0.11
Dog	4634	1101	1001	936	**883**	0.20	0.15	0.09
Horse	3679	776	783	722	**680**	0.12	0.07	-0.01
Lion	1577	435	356	338	**322**	0.26	0.22	0.18
Plane	9591	2211	2190	1957	**1789**	0.19	0.11	0.01
Map	4140	1031	847	**793**	848	0.18	0.23	0.18
Moto	5954	1391	1315	1211	**1133**	0.19	0.13	0.05
Skull	6861	1453	1358	**1210**	1298	0.11	0.17	0.07

We can see, in Fig. 3, that there exists a linear relationship between Arithmetic coding, applied to 3OT, Huffman and Arithmetic coding to DFCCE and JBIG, vs. Perimeter-8. Whereas Fig. 4 shows an exponential relationship between Efficiency and Perimeter-8. For the case of Huff-DFCCE, similar behavior appears in [15], in which Efficiency of DFCCE and Perimeter-8 were plotted, and Huffman algorithm was applied. However, the improvement now, is that the graph plotted is farther from the zero efficiency, and is given by Arith-3OT. On the other hand, compression efficiency for 3OT can be approximated by an approximated Gaussian function: $\sum_{i=1}^{3} a_i e^{(-(x-b_i)/c_i)^2}$ where a_1 = 2.2 x 10^{10}, b_1 = -75.69, c_1 = 140.5; a_2 = 2.42, b_2 = -5693, c_2 = 4226; a_3 =0.1603, b_3 = 6892, c_3 = 18840.

Observe in Fig. 4, that the trend of Huff-DFCCE suggests it will cross with the trend of JBIG, of course further than 9000 units in perimeter-8 (similar behavior was found in Sanchez-Cruz, et al, 2007 for Huff-DFCCE). However, the trend of Arith-3OT and Arith-DFCCE suggest that they never will cross with the trend of JBIG, even more, the slope of the fitted function of Arith-3OT is the smallest. This analysis allow us to say that is better to use 3OT code to represent bi-level images, whenever irregular shapes are into the images. This analysis and the trend of efficiency, suggests that for all perimeter contour coded by 3OT, in which Arithmetic coding is applied has better performance than JBIG, including for larger perimeter contour than the found in (Sanchez-Cruz, et al., 2007) in which Huffman algorithm was more effective to compress binary objects.

Of course, for each hole some extra bits are needed to represent the starting. In case of Moto shape (with the largest amount of holes) 24 holes were coded. Each code will require two starting coordinates, in the "worst case" 960 columns and 738 lines can be coded by 20 bits, multiplied by 24, gives 480bits. So, 60 extra bytes, will be required to codify the Moto shape. Obviously, not the 24 holes have the (960,738) coordinates, this amount is a "worst case", and, even though, this does not change the trends found.

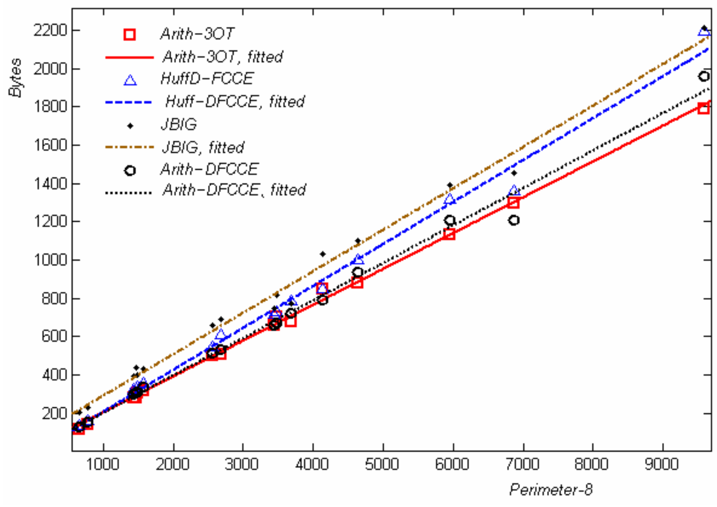


Fig. 3. Linear relationship between Arithmetic coding to 3OT, Huffman algorithm and Arithmetic coding to DFCCE and JBIG, vs. Perimeter-8

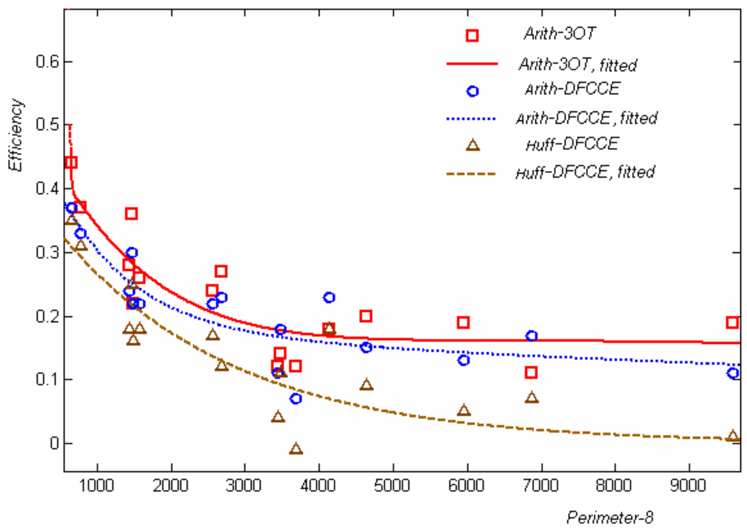


Fig. 4. Approximated functions to the obtained data

4 Conclusions and Further Work

To represent shape of binary objects, we have used the Arithmetic coding applied to different codes, we compared the results with JBIG compressor to measure their efficiency. Undoubtedly, Arithmetic applied to 3OT and DFCCE chain codes, brings better compression levels if comparing with JBIG and with Huffman applied to DFCCE, however, 81% of our sample objects were better compressed for Arithmetic to 3OT than Arithmetic to DFCCE. An interesting detailed study in this differences is suggested to be investigated between this class and the remaining19%, to see whether some common features are present. It is evident that the whole distribution follows the showed trends, which represents an improvement of a recent work in literature. So, our main contribution is to find that 3OT code constitutes best code to represent binary images with no limitation in contour perimeters. in general, about coding scheme Freeman [18] states: they "must satisfy three objectives: (1) it must faithfully preserve the information of interest; (2) it must permit compact storage and be convenient for display; and (3) it must facilitate any required processing. The three objectives are somewhat in conflict with each other, and any code necessarily involves a compromise among them”. So, we consider that 3OT has the three characteristics.

There are several methods to code a 2D object, it would also be important to compare the proposed method with other invariant algorithms in order to assess the suitability of the method in more complex scenes and real world problems.

Considering the superiority of JBIG2 over JBIG, as a future work, comparison of 3OT and DFCCE versus JBIG2 is suggested to be investigated, and also the possible application to maps, trees, text documents, and fractal images.

References

1. Mckee, J.W., Aggarwal, J.K.: Computer recognition of partial views of curved objects. IEEE Transactions on Computers C-26, 790–800 (1977)
2. Hoque, S., Sirlantzis, K., Fairhurst, M.C.: A New Chain-code Quantization Approach Enabling High Performance Handwriting Recognition based on Multi-Classifier Schemes, ICDAR. In: Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), vol. 2, p. 834 (2003)
3. Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers EC-10, 260–268 (1961)
4. Sánchez-Cruz, H.: A Proposal Method for Corner Detection with an Orthogonal Three-direction Chain Code. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 161–172. Springer, Heidelberg (2006)
5. Salem, M., Sewisy, A., Elyan, U.: A Vertex Chain Code Approach for Image Recognition. ICGST International Journal on Graphics, Vision and Image Processing 5(3) (2005)
6. Sánchez-Cruz, H., Rodríguez-Dagnino, R.M.: Compressing bi-level images by means of a 3-bit chain code. Optical Engineering 44(9), 097004 (2005)
7. Liu, Y.K., Žalik, B.: And efficient chain code with Huffman coding. Pattern Recognition 38(4), 553–557 (2005)
8. Liu, Y.K., Wei, W., Wanga, P.J., Žalik, B.: Compressed vertex chain codes. Pattern Recognition 40, 2908–2913 (2007)
9. Bribiesca, E.: A new chain code. Pattern Recognition 32(2), 235–251 (1999)
10. Rytter, W.: Compressed and fully compressed pattern matching in one and two dimensions. Proc. IEEE 88(11), 1769–1778 (2000)
11. Farach, M., Thorup, M.: String matching in Lempel-Ziv compressing strings. In: Proc. 27th Annu. Symp. Theory Computing, pp. 703–712 (1995)
12. Aguinaga, L.E., Neri-Calderón, R.A., Rodríguez-Dagnino, R.M.: Compression rates comparison of entropy coding for three-bit chain codes of bilevel images. Optical Engineering 46(8), 087007 (2007)
13. Sanchez-Cruz, H., Lopez-Cruces, M., Puga, H.: A Proposal Modification of the 3OT Chain Code. In: Thalmann, D. (ed.) Computer Graphics and Imaging, vol. 10, 300 pages, pp. 6–11. Acta Press (2008), A Publication of the International Association of Science and Technology for Development. ISBN: 978-0-88986-719-2
14. Huffman, M.: Lossless bilevel image compression. In: Sayood, K. (ed.) Lossless Compression Handbook. Academic Press, New York (2003)
15. Sanchez-Cruz, H., Bribiesca, E., Rodriguez-Dagnino, R.M.: Efficiency of chain codes to represent binary objects. Pattern Recognition 40(6), 1660–1674 (2007)
16. Howard, P., Kossentini, F., Martins, B., Forchhammer, S., Rucklidge, W.: The emerging JBIG2 standard. IEEE Transactions on Circuits and Systems for Video Technology 8(7), 838–848 (1998)
17. Ono, F., Rucklidge, W., Arps, R., Constantinescu, C.: JBIG2-the ultimate bi-level image coding standard. In: Proceedings International Conference on Image Processing, Vancouver, BC, Canada, vol. 1, pp. 140–143 (2000)
18. Freeman, H.: Computer Processing of Line-Drawing Images. ACM Computing Surveys (CSUR) 6(1), 57–97 (1974)

Finding Images with Similar Lighting Conditions in Large Photo Collections

Mauricio Díaz[1,⋆] and Peter Sturm[2]

[1] Laboratoire Jean Kuntzmann, 38400 Saint Martin d'Hères, France
[2] INRIA Grenoble Rhône-Alpes, 38330 Montbonnot, France
{Mauricio.Diaz,Peter.Sturm}@inrialpes.fr

Abstract. When we look at images taken from outdoor scenes, much of the information perceived is due to the ligthing conditions. In these scenes, the solar beams interact with the atmosphere and create a global illumination that determines the way we perceive objets in the world. Lately, exploration of the sky like the main illuminance component has began to be explored in Computer Vision. Some of these studies could be classified like color-based algorithms while some others fall in the physics-based category. However most of them assume that the photometric and geometric camera parameters are constant, or at least, that they could be determined. This work presents a simple and effective method in order to find images with similar lighting conditions. This method is based on a Gaussian mixture model of sky pixels represented by a 3D histogram in the La^*b^* color space.

1 Introduction

Nowadays, the Internet has become an interactive system that allows anyone with a compact camera to share their visual experiences. This fact has been the starting point for large online image databases such as Flickr or Picasaweb. Computer vision scientists have not taken too much time to make use of these tools, and the development of new algorithms that exploit them is an activity in progress. Among these works, an excellent example is the PhotoTourism project [16]. The main goal is to make a sparse 3D reconstruction of a popular monument from thousands of images using *structure-from-motion* type algorithms. Applications like that allow the user to explore unstructured photo collections.

The work presented in this article is also motivated by another kind of application that could exploit the richness of the photometric information available. For example in photomontage applications human intervention is often necessary to create realistic images. If one wants to insert an object from a photo into an environment determined in a second photo, both images should show up as if they had been taken under the same conditions. In that case, a human expert could retouch the images and force them to match illumination conditions and

[⋆] Supported by the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E07D402742CO.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 53–60, 2009.

shadows [1,3]. Our work is developed on the basis that we can exploit the above freely accessible collections and try to find a group of images that depict similar illumination conditions.

In a general context, one of the most successful approximations to extract information about the illumination is to use a spherical object in such a way that it can reflect and capture the light distribution when the photo is taken [17]. The main disadvantage of this method is that it requires access to the scene at the moment of the click. In his thesis [13], Love did some observations that let us think that the sky is a very important illumination source and, as consequence, one of the essential factors involved in the color perception of outdoor scenes. Some other researchers have proposed to find the behavior of the sky illumination using only images for example, using physically-based models [12], or other ones using advanced techniques for color correlation [11,10]. If one does not limit analysis to photos including sky portions, there are several works in the image completion context (*hole filling* or *inpainting*) [6,19] and in the context of color classification using raw pixels [14,18]. Some of the researchs that explore the sky as information source show restricted constraints, for example the use of a single camera or an *a priori* knowledge about the calibration (in the photometric and geometric sense). These constraints try to reduce the number of variables involved in the color perception process (surface reflectance, light sources, sensor response). For instance, the work of Lalonde *et al.* [12] proposes an algorithm to find similar sky images given the intrinsic camera parameters and a physically-based sky model. The main idea consists in finding a sky model for each given image and to compare the features that define these models. This process can only be applied to images taken with a static camera, for example time lapse sequences. In [11], an automatic photomontage system that uses a presegmented object library is described. The method used by the authors to determine the global illumination matching consists of calculating the χ^2 distance between 3D-histograms that represent the sky. Short distances are recognized as possible matches. In our experiments, this metric presents a low performance due to the high dependance to the color workspace.

The present work aims to find the matches between sky zones of multiple images in a large image collection using minimum information about the camera parameters. For this, we propose the union and improvement of three stages previously developed in other contexts. The first one, a preamble stage that allows the sky pixels segmentation. A second stage where the pixels are represented by a sum of Gaussians in the La^*b^* space (section 2.1). Finally a third stage that compares the estimated models (section 2.2). The final part of this article (section 3) presents some results and discusses the algorithm proposed.

2 Representation and Comparison of Sky Regions

2.1 Sky Model in the La^*b^* Color Space

In the literature, it is common to find physically-based models that express the sky luminance as a function of diverse variables, among them, sun position and atmospheric turbulence [9,15]. These models have been used in Computer Vision under some constraints, like a fixed point of view. In the present work the images are extracted from a large database and came from different cameras. This fact increases dramatically the problem complexity because we do not have any information about the geometry of the scene. Moreover, we do not know how the data captured by the sensors was modified during the acquisition process (most cameras apply post-processing algorithms to enhance the "perceptual quality" of the images). Our methodology is based on the camera's final output, a *jpeg* file, assuming that most of the time this image does not represent faithfully the real world illumination at the acquisition time. To accomplish the goal of this work, the first stage consists in extracting the sky pixels for each image. The application described in [8] allows us to make an automatic segmentation of the image sky.

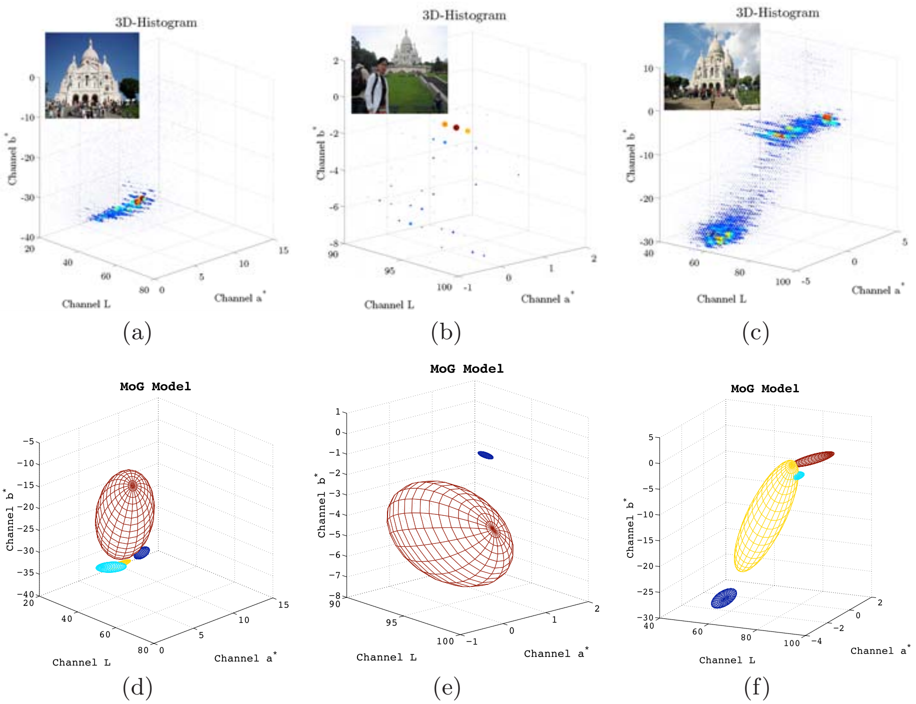
To determine a model from sky pixels is a decisive step in the formulation of our problem. For that reason, the choice of the color space plays an important role. The CIE (Commission Internationale de l'Éclairage) [4] have created several standards and nowadays, the last model published (CIECAM02) has reached an excellent performance as well as a high degree of complexity [5]. This color appereance model allows a trustworthy representation of the real world. However the parameters required are, in some cases, impossible to acquire. The La^*b^* space and the xyY space used by [11] and [12] are simpler models derived from the CIE's works. Although these color spaces are used in a vast number of works, it is important to take precautions when they are applied. For example, the spaces above mentioned are always attached to a predetermined "reference white", usually unknown. In this work we use the La^*b^* color space because of its good results in distinguishing color differences, but under the assumption that all the images were taken using a natural global illumination. That means that the "reference white" used in the transformation is the same for all images (Illuminant D65). Following the work done by Lalonde *et al.* [11], we build color histograms for the sky region in each image. These histograms are probability distributions of the colors in terms of the variables L, a^* and b^*. Figure 1 shows some of those histograms in their 3D space. Color and point size represent the bin magnitude[1]. One can note that the pixels are spread mainly on the negative part of the b^* axis which corresponds to variations between yellow and blue. On the other hand, the histograms of images, where the sky is white and/or partially cloudy, are distributed throughout the space and they are not easily distinguishable using this representation. We believe that these histograms have different modes, and that they could be modeled using a mixture of Gaussians.

According to our observations, the sky in each image is modeled following the next equation:

$$M(x) = \sum\nolimits_{k=1}^{K} \pi_{\mathbf{k}} \mathcal{N}\left(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right) \quad , \tag{1}$$

for an undetermined K (in our implementation we limit this value to 4). One well-known method to find the parameters in equation (1) is the Expectation-Maximization algorithm (EM) [2]. The output of this method corresponds to the

[1] For figures, the reader is kindly invited so see the electronic version.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Sky 3D-histograms and ellipsoids for the corresponding MoG models. (a) Image with sunny sky, (b) completely cloudy sky, (c) partially cloudy sky. (d) Model with four Gaussians. (e) Model with two Gaussians. (f) Model with four Gaussians.

variables π_k, $\mu_{\mathbf{k}}$, $\mathbf{\Sigma_k}$ that describe the model in such a way that the probability of the solution is maximized. Our model is formulated in terms of a joint probability distribution of the sky pixels X, the latent variables Z and the goal is to find the set of values θ that maximize the likelihood function (θ represents the variables $\pi_{\mathbf{k}}$, $\mu_{\mathbf{k}}$ and $\mathbf{\Sigma_k}$). The values La^*b^* of the sky pixels $\mathbf{x_n}$ are organized in the matrix X in which the n^{th} row is given by $\mathbf{x_n^{\mathsf{T}}}$. In order to briefly summarize this algorithm, an iterative 2 step process runs until it reaches a convergence parameter. The **E** step calculates the joint probability distribution $P(\mathsf{Z}|\mathsf{X},\theta^{\text{last}})$:

$$P(\mathsf{Z}|\mathsf{X},\theta^{\text{last}}) = \frac{\pi_{\mathbf{k}}\mathcal{N}\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K}\pi_{\mathbf{j}}\mathcal{N}\left(\mathbf{x_n}|\mu_{\mathbf{j}},\mathbf{\Sigma_j}\right)} ,$$

and the **M** step updates the values of $\pi_{\mathbf{k}}$, $\mu_{\mathbf{k}}$, $\mathbf{\Sigma_k}$. The model dimension is a crucial factor. In our case, the number of Gaussians used in the mixture is determined based on the Akaike information criterion (AIC) [2]. Different values of K in the equation (1) are evaluated and, the model that maximizes the likelihood is chosen. In figure 1, the ellipsoids that form the Gaussian Mixture model are shown for a constant variance.

2.2 Comparison between Histograms of the Sky

Once the sky model for each image is estimated, we proceed to compare different models. In the present case, it is necessary to measure the difference between two or more probability distributions. In our context, the well-known Kullback-Leibler divergence (KL) should be a good option, although it does not possess the property of symmetry. Given two probability distributions $p(x)$ and $q(x)$ the KL divergence is defined by:

$$\mathsf{KL}(p||q) = -\int p(x) ln \left\{ \frac{q(x)}{p(x)} \right\} dx \ . \tag{2}$$

It is widely proven that when the distributions are Gaussian, equation (2) can be expressed in closed-form. However, in the case of a Gaussian mixture, it is difficult to find an analytically tractable expression or even more, a computer algorithm to solve this problem efficiently. According to the work of Hershey and Olsen [7], the only method for estimating $\mathsf{KL}(p||q)$ with arbitrary accuracy when $p(x)$ and $q(x)$ are mixtures of Gaussians, is the Monte Carlo simulation. Nevertheless, other approximations may be valid, depending on the context. For example, a commonly used approximation is the simplification of the Gaussian mixtures $p(x)$ and $q(x)$ by simple Gaussians $\tilde{p}(x)$ and $\tilde{q}(x)$. In this case, the mean and covariance estimated are:

$$\begin{aligned} \mu_{\tilde{p}} &= \sum\nolimits_a \pi_a \mu_a \\ \Sigma_{\tilde{p}} &= \sum\nolimits_a \pi_a \left(\Sigma_a + (\mu_a - \mu_{\tilde{p}})(\mu_a - \mu_{\tilde{p}})^{\mathsf{T}} \right) \ . \end{aligned} \tag{3}$$

The KL divergence ($\mathsf{KL}_{\mathrm{sim}}$) is calculated using the estimated mean ($\mu_{\tilde{p}}$) and variance ($\Sigma_{\tilde{p}}$). Hershey and Olsen use variational methods to find a better approximation of the KL divergence. One of their contributions is a measure that satisfies the symmetry property but not the property of positivity. In this case, the approximated divergence $\mathsf{KL}_{\mathrm{app}}(p||q)$ is given by:

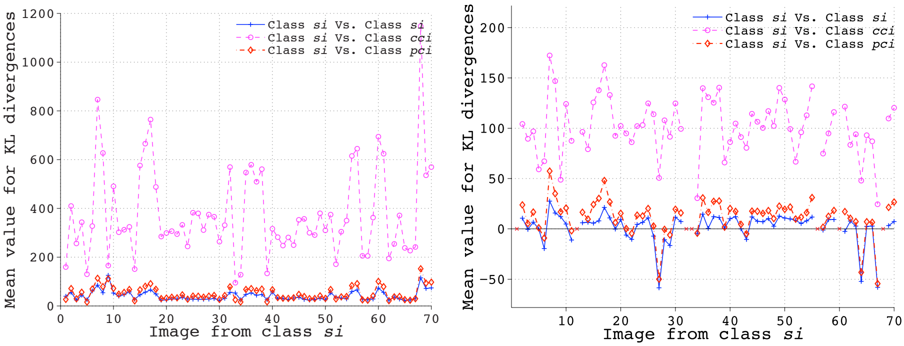
$$\mathsf{KL}_{\mathrm{app}}(p||q) = \sum_a \pi_a \log \frac{\sum_{a'} \pi_{a'} e^{-\mathsf{KL}(p||p')}}{\sum_b \omega_b e^{-\mathsf{KL}(p||q)}} \ . \tag{4}$$

This value could be seen as a measure of dissimilarity between two distributions. Hershey and Olsen's contribution allows us to compare two sky models keeping the variables for each Gaussian that composes the mixture.

3 Results

This section shows some results of the experiments developed using the models estimated and the comparison measure described in the last section. To test our method, we re-create a database with 4250 images of Sacre Cœur's Cathedral (Paris) downloaded from the Flickr website[2]. Images that do not correspond to

[2] http://www.flickr.com/



(a) (b)

Fig. 2. Mean values for the intra-class and inter-class divergence, (a) using the $\mathsf{KL}_{\text{simple}}$ divergence and (b) using the KL_{app} divergence

Table 1. Average of the inter and intra-class divergences for all images in the class *si*

	Average $\mathsf{KL}_{\text{simple}}$	Average KL_{app}
si class	45.60	4.27
pci class	351.63	97.87
cci class	50.95	13.73

the desired scenes or are taken during the night or using artificial lighting had been removed.

Our goal is to compare two or more outdoor images using the above methods applied on sky regions. To test our approach, we classified 500 images from the database by hand according to: sunny (*si*), partially cloudy (*pci*) and completely cloudy (*cci*). It is important to emphasize the subjective nature of this “ground truth”. The main idea consists in comparing the distances between one selected image and the other ones that belong to the same class (intra-class distance) and those belonging to other classes (inter-class distance). Figure 2 shows the intra-class and inter-class mean distances computed for 70 images from the class *si* using the KL_{sim} divergence and the KL_{app} divergence. For these two measures, we find that the class *cci* differs clearly from the other two by medium values of greater amplitude. However, among the class *si* and the class *pci* the difference is not sufficient when using the measure KL_{sim}. As we may expect, the approximated KL_{app} divergence shows a better performance and discrimination is easier. Table 1 presents the average of the inter and intra-class divergences for all images in the class *si* according to the two measures described. The difference between images belonging to classes *si* and *cci* is clearer when we use the KL_{app} divergence. Once again, it can be explained by a better approximation of the KL divergence.

Another experiment was developed in order to find the success rate of the algorithm when running on 500 images using the KL_{app} divergence. The objective this time is to find the closest images and whether or not they belong to the

Table 2. Success rate on 500 images

Images	1^{st} Image	2^{nd} Image	3^{rd} Image	4^{th} Image
All	78.7%	75.4%	72.0%	71.2%
Class *cci*	77.3%	74.6%	70.8%	68.7%
Class *si*	64.0%	68.8%	64.8%	59.1%
Class *pci*	93.2%	82.0%	79.9%	84.8%

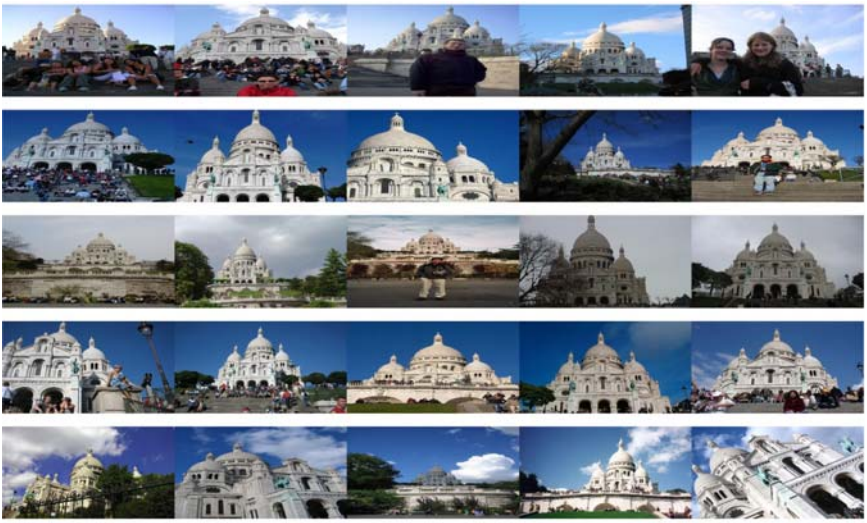


Fig. 3. Subset of typical results. In the first column we observe the query image. The following columns from left to right, show the images closest to the query image.

same class that the query image, according to the ground truth (see table 2). Figure 3 shows the four most similar images found for three query images.

4 Conclusions

In this paper, we proposed a 3-steps pipeline (segmentation, modelisation and comparison) for grouping outdoor images with similar lighting conditions, based on techniques previously developed. The model choosen for the sky pixles has been a mixture of Gaussian, based on the observations of histograms with multiple modes. This distribution allows us to accurately model the pixels, especially for images where the sky is not of uniform color. On the other hand, the computation of the KL divergence has proven to be an adequate tool to group similar skies, despite the approximations made in the operation. These approximations must be carefully chosen in order to keep the benefits of using a multimodal distribution as model. For a finer grouping, other factors may be considered such as the position of the sun, clouds, shadows or information about the camera parameters that could be extracted from the metadata. Also, to obtain credible photomontages, information about the geometry of the scene could be valuable. The inclusion of these variables might produce natural and realistic compositions.

References

1. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Interactive digital photomontage. In: SIGGRAPH 2004, pp. 294–302. ACM, New York (2004)
2. Bishop, C.M.: Pattern recognition and machine learning (information science and statistics). Springer, Heidelberg (August 2006)
3. Burt, P.J., Kolczynski, R.J.: Enhanced image capture through fusion. In: ICCV 1993, pp. 173–182 (1993)
4. Committee, C.T.: Spatial distribution of daylight - luminance distributions of various reference skies., Tech. Report CIE-110-1994, Commission Internationale de l'Éclairage, CIE (1994)
5. Committee, C.T.: Colour appearance model for colour management applications, Tech. Report CIE-TC8-01, Commission Internationale de l'Éclairage, CIE (2002)
6. Hays, J., Efros, A.: Scene completion using millions of photographs. In: SIGGRAPH 2007, vol. 26(3,4) (2007)
7. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: ICASSP 2007, vol. 4, pp. IV–317–IV–320 (2007)
8. Hoiem, D., Efros, A., Hebert, M.: Geometric context from a single image. In: ICCV 2005, pp. 654–661 (2005)
9. Igawa, N., Koga, Y., Matsuzawa, T., Nakamura, H.: Models of sky radiance distribution and sky luminance distribution. Solar Energy 77(2), 137–157 (2004)
10. Jacobs, N., Roman, N., Pless, R.: Consistent temporal variations in many outdoor scenes. In: CVPR 2007, pp. 1–6 (2007)
11. Lalonde, J., Hoiem, D., Efros, A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. In: SIGGRAPH 2007, vol. 26 (2007)
12. Lalonde, J., Narasimhan, S.G., Efros, A.: What does the sky tell us about the camera? In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 354–367. Springer, Heidelberg (2008)
13. Love, R.C.: Surface reflection model estimation from naturally illuminated image sequences, Tech. report, The University of Leeds, Ph.D. thesis (1997)
14. Manduchi, R.: Learning outdoor color classification. IEEE Transactions on PAMI 28(11), 1713–1723 (2006)
15. Perez, R., Seals, R., Michalsky, J.: All-weather model for sky luminance distribution - preliminary configuration and validation. Solar Energy 50(3), 235–245 (1993)
16. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vision 80(2), 189–210 (2008)
17. Stumpfel, J., Jones, A., Wenger, A., Tchou, C., Hawkins, T., Debevec, P.: Direct hdr capture of the sun and sky. In: AFRIGRAPH 2004, pp. 145–149 (2004)
18. Sunkavalli, K., Romeiro, F., Matusik, W., Zickler, Y., Pfister, H.: What do color changes reveal about an outdoor scene? In: CVPR 2008, pp. 1–8 (2008)
19. Wilczkowiak, M., Brostow, G.J., Tordoff, B., Cipolla, R.: Hole filling through photomontage. In: BMVC 2005, July 2005, pp. 492–501 (2005)

Color Image Registration under Illumination Changes*

Raúl Montoliu[1], Pedro Latorre Carmona[2], and Filiberto Pla[2]

[1] Dept. Arquitectura y Ciéncias de los Computadores
[2] Dept. Lenguajes y Sistemas Informáticos.
Jaume I University
Campus Riu Sec s/n 12071 Castellón, Spain
{montoliu,latorre,pla}@uji.es
http://www.vision.uji.es

Abstract. The estimation of parametric global motion has had a significant attention during the last two decades, but despite the great efforts invested, there are still open issues. One of the most important ones is related to the ability to simultaneously cope with viewpoint and illumination changes while keeping the method accurate. In this paper, a Generalized least squared-based motion estimator model able to cope with large geometric transformations and illumination changes is presented. Experiments are made on a series of images showing that the presented technique provides accurate estimates of the motion and illumination parameters.

1 Introduction

Image registration could be defined as the process to transform an image to match another image as correctly as possible. This process is necessary when we want to compare or to integrate the data information from the two images (see [18] for a review of image registration methods). During image acquisition of a scene, many factors intervene: the position and the distance from the camera (or sensor) to the scene, the illumination, the nature of the objects to be imaged, etc. Any change in these factors implies that the data in the corresponding images are not directly comparable. During the last few years, a special interest has emerged in relation to the need to cope with simultaneous viewpoint and illumination changes ([11], [13], [1], [2], to cite a few works). Interesting examples could be found in image databases where we can obtain images of the same place acquired with different cameras and with different acquisition conditions.

In general, the *direct geometric registration problem* can be solved minimizing an error function in relation to the difference in the pixel values between an image that may be called *Test image* and the *Reference image*. In particular, it can be formally written as:

$$\min_{g} \sum_{\mathbf{q}\in\mathbb{R}} \|\mathcal{I}_1(\mathbf{q_i}) - \mathcal{I}_2(\mathcal{G}(\mathbf{q_i};\mathbf{g}))\|^2 \tag{1}$$

* This work has been partially funded by projects AYA2008-05965-C04-04 and CSD2007-00018 from Spanish Ministry of Science and Innovation.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 61–68, 2009.

where $\mathcal{I}_1$ and $\mathcal{I}_2$ are two input images, $\mathbf{q_i} = (x_i, y_i)^T$ are the pixel coordinates, $\mathbf{g}$ is the vector of motion parameters and $\mathcal{G}$ is the function to transform the pixel coordinates from one image to the other. The function $\mathcal{G}$ is expressed, for instance, in an affine motion (Eq. 2) as follows:

$$\mathcal{G}(\mathbf{q_i};\mathbf{g}) = \begin{pmatrix} a_1 x_i + b_1 y_i + c_1 \\ a_2 x_i + b_2 y_i + c_2 \end{pmatrix}, \mathbf{g} = (a_1, b_1, c_1, a_2, b_2, c_2). \tag{2}$$

If photometric changes are also considered, these may be modeled by a transformation $\mathcal{P}$ with parameter vector $\mathbf{p}$ and the minimization would therefore be:

$$\min_{g} \sum_{\mathbf{q}\in\mathbb{R}} \|\mathcal{I}_1(\mathbf{q_i}) - \mathcal{P}(\mathcal{I}_2(\mathcal{G}(\mathbf{q_i};\mathbf{g}));\mathbf{p})\|^2 \tag{3}$$

To solve the problem shown in Eq. 3, Bartoli developed the *Dual Inverse Compositional* (DIC) estimation technique [2] considering Eq. 3 and then applying an inverse compositional update rule for both the geometric and photometric transformations. See [2] for details on the steps of the algorithm used to assess the geometric registration and illumination compensation parameters.

In this paper a generalized least squares-based non-linear motion estimation technique that incorporates the capability to deal with color illumination changes is presented (it will be called the GLSIC method), where illumination changes are modeled considering an affine transformation framework. The method is based on the Generalized Least Squares (GLS) motion estimation method introduced by Montoliu and Pla in [12], and where a new criterion function is proposed, incorporating these illumination changes. The GLS method is applied on this criterion function, deriving a new set of equations whose solutions allow the simultaneous assessment of the geometric and affine illumination transformation parameters. It is shown that the proposed method provides better results than the method recently described in [2]. This method is used since it is, for the best of our knowledge, the most relevant technique that simultaneously estimates the motion and the illumination transformation parameters in color images.

The rest of the paper is organized as follows: Section 2 justifies the use of a complete affine transform to model illumination changes. In Section 3, the GLS for general problems is briefly introduced. Section 4 presents the GLSIC method. Section 5 shows the experiments and results obtained by the proposed method. Conclusions are drawn in Section 6.

2 Illumination Compensation Model

Illumination compensation is closely related to *chromatic adaptation* in human colour vision. The first chromatic adaptation experiments started in the late 1940s. A few years later, Wyszecki and Stiles proved in a *human asymmetric matching* experiment [16] that a diagonal linear matrix transform would be enough to reproduce the experiment of asymmetric matching. However, West and Brill [15] and others proved later that for a given set of sensor sensitivities a diagonal transform could only cover a restricted group of object colours and illuminant spectra. Finlayson et al [3] argued that a diagonal transform would be enough for the modeling of an illumination change if the camera has extremely narrow-band sensors, which is often not the case. There are other cases where diagonal illumination compensation can fail, for instance, if there are other processes happening like bias in the camera, or saturated colours in the scene. In the latter case, some colours would *fall* in (and outside of) the camera gamut boundary [4]. This is the reason why the use of a complete (full) affine transform in the form $\mathbf{\Omega} \cdot \mathcal{I}(\mathbf{q}) + \mathbf{\Phi}$ is justified (see, for example, [6], [9], [17], to cite a few), where $\mathbf{\Omega} \in \mathbb{R}^{3\times 3}$ is a full matrix, with elements ω_{kl}, $(k, l = 1, \ldots, 3)$, and $\mathbf{\Phi} \in \mathbb{R}^3$, a vector with elements ϕ_k, $(k = 1, \ldots, 3)$.

3 Generalized Least Squares Estimation for General Problems

In general, the GLS estimation problem can be expressed as follows (see [12] for more details):

$$\text{minimize } \{\Theta_\upsilon = \upsilon^T \upsilon\} \text{ subject to } \xi(\chi, \gamma) = 0, \tag{4}$$

where:

- υ is a vector of r unknown residuals in the observation space, that is, $\upsilon = \gamma - \tilde{\gamma}$, where γ and $\tilde{\gamma}$ are the unperturbed and actually measured vector of observations, respectively.
- $\chi = (\chi^1, \ldots, \chi^m)^T$ is a vector of m parameters;
- γ formed by r elements γ_i, $\gamma = (\gamma_1, \ldots, \gamma_r)^T$, each one is an observation vector with n components $\gamma_i = (\gamma_i^1, \ldots, \gamma_i^n)^T$
- $\xi(\chi, \gamma)$ formed by r elements $\xi_i(\chi, \gamma_i)$, $\xi(\chi, \gamma) = (\xi_1(\chi, \gamma_1), \ldots, \xi_r(\chi, \gamma_r))^T$, each one is, in general, a set of f functions that depend on the common vector of parameters χ and on an observation vector γ_i, $\xi_i(\chi, \gamma_i) = (\xi_i^1(\chi, \gamma_i), \ldots, \xi_i^f(\chi, \gamma_i))^T$. Those functions can be non-linear.

Thus, the solution of (4) can be addressed as an iterative optimization process starting with an initial guess of the parameters $\widehat{\chi}(0)$. At each iteration j, the algorithm estimates $\widehat{\Delta\chi}(j)$ to update the parameters as follows: $\widehat{\chi}(j) = \widehat{\chi}(j-1) + \widehat{\Delta\chi}(j)$. The process is stopped if the improvement $\widehat{\Delta\chi}(j)$ is lower than a threshold. The improvement $\widehat{\Delta\chi}(j)$ can be expressed as follows:

$$\widehat{\Delta\chi}(j) = (\mathbf{A}^T \mathbf{Q} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Q} \mathbf{e}, \tag{5}$$

where $\mathbf{Q} = (\mathbf{B}\mathbf{B}^T)^{-1}$. Equation 5 can also be expressed as:

$$\widehat{\Delta\chi}(j) = \left(\sum_{i=1\ldots r} \mathbf{N}_i \right)^{-1} \left(\sum_{i=1\ldots r} \mathbf{T}_i, \right), \tag{6}$$

with $\mathbf{N}_i = \mathbf{A}_i^t (\mathbf{B}_i \mathbf{B}_i^t)^{-1} \mathbf{A}_i$ and $\mathbf{T}_i = \mathbf{A}_i^t (\mathbf{B}_i \mathbf{B}_i^t)^{-1} \mathbf{e}_i$, where $\mathbf{B}_i$ is an $\mathbb{R}^{f\times n}$ matrix with elementes $b_i(kl) = \frac{\partial \xi_i^k(\widehat{\chi}(j-1), \gamma_i)}{\partial \gamma_i^l}$ $(k = 1, \ldots, f;\ l = 1, \ldots, n)$; $\mathbf{A}_i$ is an $\mathbb{R}^{f\times m}$ matrix with elements $a_i(kl) = \frac{\partial \xi_i^k(\widehat{\chi}(j-1), \gamma_i)}{\partial \chi^l}$ $(k = 1, \ldots, f;\ l = 1, \ldots, m)$; and finally $\mathbf{e}_i$ is an $\mathbb{R}^f$ vector with elements $e_i(k) = -\xi_i^k(\widehat{\chi}(j-1), \gamma_i)$ $(k = 1, \ldots, f)$.

Table 1. $\mathbf{A}_i$ matrix for affine motion. First part.

Function	∂a_1	∂b_1	∂c_1	∂a_2	∂b_2	∂c_2
$\xi_1(\chi,\gamma_i)$	$x_i R_2^x$	$y_i R_2^x$	R_2^x	$x_i R_2^y$	$y_i R_2^y$	R_2^y
$\xi_2(\chi,\gamma_i)$	$x_i G_2^x$	$y_i G_2^x$	G_2^x	$x_i G_2^y$	$y_i G_2^y$	G_2^y
$\xi_3(\chi,\gamma_i)$	$x_i B_2^x$	$y_i B_2^x$	B_2^x	$x_i B_2^y$	$y_i B_2^y$	B_2^y

Table 2. $\mathbf{A}_i$ matrix for affine and projective motion. Second part.

Function	$\partial\alpha_{11}$	$\partial\alpha_{12}$	$\partial\alpha_{13}$	$\partial\alpha_{21}$	$\partial\alpha_{22}$	$\partial\alpha_{23}$	$\partial\alpha_{31}$	$\partial\alpha_{32}$	$\partial\alpha_{33}$	$\partial\beta_1$	$\partial\beta_2$	$\partial\beta_3$
$\xi_1(\chi,\gamma_i)$	$-R_1$	$-G_1$	$-B_1$	0	0	0	0	0	0	-1	0	0
$\xi_2(\chi,\gamma_i)$	0	0	0	$-R_1$	$-G_1$	$-B_1$	0	0	0	0	-1	0
$\xi_3(\chi,\gamma_i)$	0	0	0	0	0	0	$-R_1$	$-G_1$	$-B_1$	0	0	-1

4 GLS-Based Color Motion Estimation under Illumination Changes

In the GLSIC formulation of the motion estimation problem, the function $\xi_i(\chi,\gamma_i)$ is expressed as follows:

$$\xi_i(\chi,\gamma_i) = \mathcal{I}_2(\mathbf{q}_i') - \mathcal{P}^{-1}(\mathcal{I}_1(\mathbf{q}_i);\mathbf{p}) \tag{7}$$

with $\mathcal{I}_1(\mathbf{q}_i) = (R_1(\mathbf{q}_i), G_1(\mathbf{q}_i), B_1(\mathbf{q}_i))^T$ and $\mathcal{I}_2(\mathbf{q}_i') = (R_2(\mathbf{q}_i'), G_2(\mathbf{q}_i'), B_2(\mathbf{q}_i'))^T$, where $\mathbf{q}_i'$ has been introduced to simplify notation as: $\mathbf{q}_i' = \mathcal{G}(\mathbf{q}_i;\mathbf{g})$. Note that in this case the number of functions f is 3. Eq. 7 can also be writen in a more convenient way as follows:

$$\begin{aligned}
\xi_i^1(\chi,\gamma_i) &= R_2(\mathbf{q}_i') - (R_1(\mathbf{q}_i)\omega_{11} + G_1(\mathbf{q}_i)\omega_{12} + B_1(\mathbf{q}_i)\omega_{13} + \phi_1)\\
\xi_i^2(\chi,\gamma_i) &= G_2(\mathbf{q}_i') - (R_1(\mathbf{q}_i)\omega_{21} + G_1(\mathbf{q}_i)\omega_{22} + B_1(\mathbf{q}_i)\omega_{23} + \phi_2)\\
\xi_i^3(\chi,\gamma_i) &= B_2(\mathbf{q}_i') - (R_1(\mathbf{q}_i)\omega_{31} + G_1(\mathbf{q}_i)\omega_{32} + B_1(\mathbf{q}_i)\omega_{33} + \phi_3)
\end{aligned} \tag{8}$$

where $R_1(\mathbf{q}_i)$, $G_1(\mathbf{q}_i)$ and $B_1(\mathbf{q}_i)$ are the R, G and B, components of the first color image in the sequence (*Reference image*) at the point $\mathbf{q}_i$, and $R_2(\mathbf{q}_i')$, $G_2(\mathbf{q}_i')$ and $B_2(\mathbf{q}_i')$ are the R, G and B, components of the second color image in the sequence (*Test image*) at the transformed point $\mathbf{q}_i' = \mathcal{G}(\mathbf{q}_i;\mathbf{g})$. In this case, each observation vector γ_i is related to each pixel $\mathbf{q}_i$, with r being the number of pixels in the area of interest.

Let us define the observation vector as $\gamma_i = (R_1(\mathbf{q}_i), G_1(\mathbf{q}_i), B_1(\mathbf{q}_i), x_i, y_i)$. The vector of parameters is defined as follows: $\chi = (\mathbf{g}, \mathbf{p})^T$. Due to the high dimensionality of the parameter vector it is difficult to describe $\mathbf{A}_i$, $\mathbf{B}_i$ using matrix form. Tables will be used instead. For affine motion, $\mathbf{A}_i$ is shown in Tables 1 and 2; $\mathbf{B}_i$ is shown in Tables 3 and 4. For projective motion, $\mathbf{A}_i$ is shown in Tables 5 and 2; $\mathbf{B}_i$ is shown in Tables 3 and 6.

In Tables 1 to 6, R_1^x, R_1^y, G_1^x, G_1^y, B_1^x, B_1^y, R_2^x, R_2^y, , G_2^x, G_2^y, B_2^x and B_2^y have been introduced to simplify notation as follows: $R_1^x(\mathbf{q}_i)$, $R_1^y(\mathbf{q}_i)$, $G_1^x(\mathbf{q}_i)$, $G_1^y(\mathbf{q}_i)$, $B_1^x(\mathbf{q}_i)$, $B_1^y(\mathbf{q}_i)$ (components of the gradient of the R, G, B bands of the reference image at

Table 3. $\mathbf{B}_i$ matrix for affine and projective motion. First part.

Function	∂R_1	∂G_1	∂B_1
$\xi_1(\chi, \gamma_i)$	$-\alpha_{11}$	$-\alpha_{12}$	$-\alpha_{13}$
$\xi_2(\chi, \gamma_i)$	$-\alpha_{21}$	$-\alpha_{22}$	$-\alpha_{23}$
$\xi_3(\chi, \gamma_i)$	$-\alpha_{31}$	$-\alpha_{32}$	$-\alpha_{33}$

Table 4. $\mathbf{B}_i$ matrix for affine motion. Second part.

Function	∂x	∂y
$\xi_1(\chi, \gamma_i)$	$(a_1 R_2^x + a_2 R_2^y) - (\alpha_{11} R_1^x + \alpha_{12} G_1^x + \alpha_{13} B_1^x)$	$(b_1 R_2^x + b_2 R_2^y) - (\alpha_{11} R_1^y + \alpha_{12} G_1^y + \alpha_{13} B_1^y)$
$\xi_2(\chi, \gamma_i)$	$(a_1 G_2^x + a_2 G_2^y) - (\alpha_{21} R_1^x + \alpha_{22} G_1^x + \alpha_{23} B_1^x)$	$(b_1 G_2^x + b_2 G_2^y) - (\alpha_{21} R_1^y + \alpha_{22} G_1^y + \alpha_{23} B_1^y)$
$\xi_3(\chi, \gamma_i)$	$(a_1 B_2^x + a_2 B_2^y) - (\alpha_{31} R_1^x + \alpha_{32} G_1^x + \alpha_{33} B_1^x)$	$(b_1 B_2^x + b_2 B_2^y) - (\alpha_{31} R_1^y + \alpha_{32} G_1^y + \alpha_{33} B_1^y)$

```
Input: Images $I_1 = (R_1, G_1, B_1)^T$ and $I_2 = (R_2, G_2, B_2)^T$
Output: $\widehat{\chi}$, the vector of estimated motion parameters.
Calculate image gradients.
$j = 0$.
Set $\Omega(0) = \mathbf{I}$, $\Phi(0) = (0, 0, 0)^T$ and $\mathbf{g}(0) = \text{FeatureStep}(I_1, I_2)$.
$\widehat{\chi}(0) = (\mathbf{g}(0), \mathbf{p}(0))^T$, with $\mathbf{p}(0) = (\omega_{11}(0), \ldots, \omega_{33}(0), \phi_1(0), \ldots, \phi_3(0))$.
repeat
    $j = j + 1$.
    Update matrices $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{e}_i$ using $\widehat{\chi}(j-1)$.
    Estimate $\widehat{\Delta\chi}(j)$.
    $\widehat{\chi}(j) = \widehat{\chi}(j-1) + \widehat{\Delta\chi}(j)$.
until $|\widehat{\Delta\chi}(j)|$ is small enough.
$\widehat{\chi} = \widehat{\chi}(j)$.
```

Algorithm 1. The GLSIC algorithm

point $\mathbf{q}_i$), $R_2^x(\mathbf{q}_i')$, $R_2^y(\mathbf{q}_i')$, $G_2^x(\mathbf{q}_i')$, $G_2^y(\mathbf{q}_i')$, $B_2^x(\mathbf{q}_i')$ and $B_2^y(\mathbf{q}_i')$ (components of the gradient of the R,G,B bands of the test image at point $\mathbf{q}_i'$), respectively.

In addition, N_d, N_1, N_2, N_3, N_4, N_5 and N_6 would be:

$$N_d = (dx_i + ey_i + 1), N_1 = a_1 x_i + b_1 y_i + c_1, N_2 = a_2 x_i + b_2 y_i + c_2$$
$$N_3 = \frac{a_1 N_d - dN_1}{N_d^2}, N_4 = \frac{a_2 N_d - dN_2}{N_d^2}, N_5 = \frac{b_1 N_d - eN_1}{N_d^2}, N_6 = \frac{b_2 N_d - eN_2}{N_d^2} \tag{9}$$

The estimation process is summarized in Algorithm 1. A Feature-based Step is used to initialize the motion estimator (whenever the deformation between images is quite large a good initial vector of motion parameters is needed). It mainly consists of a SIFT-based technique [10] to detect and describe interest points, where for each interest point belonging to the first image a *K-NN* search strategy is performed to find the k-closest interest points in the second image. Finally, for estimating the first approximation of the motion parameters a random sampling technique is used [14].

Regarding the illumination parameters at $\widehat{\chi}(0)$, they have initially been set to: $\Omega = \mathbf{I}$ and $\Phi = (0, 0, 0)^T$.

Table 5. $\mathbf{A}_i$ matrix for projective motion. First part.

Function	∂a_1	∂b_1	∂c_1	∂a_2	∂b_2	∂c_2	∂d	∂e
$\xi_1(\chi, \gamma_i)$	$\frac{xR_2^x}{N_d}$	$\frac{yR_2^x}{N_d}$	$\frac{R_2^x}{N_d}$	$\frac{xR_2^y}{N_d}$	$\frac{yR_2^y}{N_d}$	$\frac{R_2^y}{N_d}$	$\frac{-x_i R_2^x N_1 - x_i R_2^y N_2}{N_d^2}$	$\frac{-y_i R_2^x N_1 - y_i R_2^y N_2}{N_d^2}$
$\xi_2(\chi, \gamma_i)$	$\frac{xG_2^x}{N_d}$	$\frac{yG_2^x}{N_d}$	$\frac{G_2^x}{N_d}$	$\frac{G_2^y}{N_d}$	$\frac{yG_2^y}{N_d}$	$\frac{G_2^y}{N_d}$	$\frac{-x_i G_2^x N_1 - x_i G_2^y N_2}{N_d^2}$	$\frac{-y_i G_2^x N_1 - y_i G_2^y N_2}{N_d^2}$
$\xi_3(\chi, \gamma_i)$	$\frac{xB_2^x}{N_d}$	$\frac{yB_2^x}{N_d}$	$\frac{B_2^x}{N_d}$	$\frac{xB_2^y}{N_d}$	$\frac{yB_2^y}{N_d}$	$\frac{B_2^y}{N_d}$	$\frac{-x_i B_2^x N_1 - x_i B_2^y N_2}{N_d^2}$	$\frac{-y_i B_2^x N_1 - y_i B_2^y N_2}{N_d^2}$

Table 6. $\mathbf{B}_i$ matrix for proyective motion. Second part.

Function	∂x	∂y
$\xi_1(\chi, \gamma_i)$	$(N_3 R_2^x + N_4 R_2^y) - (\alpha_{11} R_1^x + \alpha_{12} G_1^x + \alpha_{13} B_1^x)$	$(N_5 R_2^x + N_6 R_2^y) - (\alpha_{11} R_1^y + \alpha_{12} G_1^y + \alpha_{13} B_1^y)$
$\xi_2(\chi, \gamma_i)$	$(N_3 G_2^x + N_4 G_2^y) - (\alpha_{21} R_1^x + \alpha_{22} G_1^x + \alpha_{23} B_1^x)$	$(N_5 G_2^x + N_6 G_2^y) - (\alpha_{21} R_1^y + \alpha_{22} G_1^y + \alpha_{23} B_1^y)$
$\xi_3(\chi, \gamma_i)$	$(N_3 B_2^x + N_4 B_2^y) - (\alpha_{31} R_1^x + \alpha_{32} G_1^x + \alpha_{33} B_1^x)$	$(N_5 B_2^x + N_6 B_2^y) - (\alpha_{31} R_1^y + \alpha_{32} G_1^y + \alpha_{33} B_1^y)$

Table 7. Results of the registration using the four similarity measures

	NAAE	NCC	ISC	SCC
GLSIC	0.9469	0.9662	0.7499	0.9648
DIC	0.9341	0.9515	0.7204	0.9528

5 Experiments and Results

In order to test the accuracy of the proposed motion estimation technique, several experiments were performed using a set of challenging images (some of them are shown in Fig. 1) obtained from several sources, including: Bartoli's example[1], Brainard's examples[2], Simon Fraser University Computational Vision Lab's examples[3] and Oxford's Visual Geometry Group's examples[4]. The first three images in the first row of Figure 1 were acquired by ourselves using a conventional digital camera and varying the illumination conditions. In all image pairs there exists a simultaneous geometric and photometric transformation. The GLSIC method was tested against the DIC method [2]. For each image pair, first, the Feature-based step was performed to obtain a good inital motion parameters vector. Then, both algorithms were applied using this initialization, obtaining two output parameters χ_{GLSIC} and χ_{DIC}. With the estimated parameters, the *Test image* can be geometrically and photometrically transformed. Then if the parameters have been correctly estimated, the resulting images ($\mathcal{I}_{GLSIC}$ and $\mathcal{I}_{DIC}$) should be very similar to the corresponding reference images.

Figure 2 shows the results obtained with the proposed technique for Bartoli's image, used in [2]. The first two images are the *Test* and the *Reference* image. The third is a panoramic image with the result of the registration. Note how the motion and the illumination parameters have been correctly estimated.

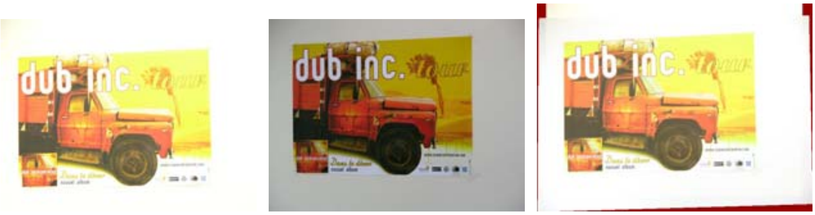
[1] http://www.lasmea.univ-bpclermont.fr/Personnel/Adrien.Bartoli/Research/DirectImageRegistration/index.html

[2] http://color.psych.upenn.edu/brainard/

[3] http://www.cs.sfu.ca/c̃olour/data/objects_under_different_lights/index.html

[4] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

Fig. 1. Some of the images used in the experiments



(a) Test Image (b) Reference Image (c) Panoramic Image

Fig. 2. Registration result using Bartoli's Example in [2]

Four image similarity measures were used to assess the quality of the estimation: the Normalized Correlation Coefficient (NCC [5]), the Increment Sign Correlation coefficient (ISC [7]), the Selective Correlation Coefficient (SCC [8]) and the Normalized Average of Absolute Errors ($NAAE$) defined as:

$$NAAE(aae) = \begin{cases} 0 & \text{if } aae > TH \\ \frac{TH-aae}{TH} & \text{otherwise} \end{cases}, \tag{10}$$

where aae is the average of the absolute errors for all the corresponding pixels in the images, and TH is a constant. The four measures produce values from 0 (low similarity) to 1 (high similarity). Table 7 shows the average of the values obtained for all experiments. Note how the proposed estimation technique overcomes Bartoli's method [2] for all similarity measures.

6 Conclusions

In this paper, a new method able to assess the geometric and photometric transformation between image pairs has been presented. It uses a Generalized Least Squares estimation framework combined with an affine transform illumination model. It has been tested in a series of images from different sources overcoming what is considered the reference method for registration with color illumination compensation [2].

References

1. Bartoli, A.: Groupwise geometric and photometric direct image registration. In: Proceedings of the British Machine Vision Conference, pp. 157–166 (2006)
2. Bartoli, A.: Groupwise geometric and photometric direct image registration. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 2098–2108 (2008)
3. Finlayson, G.D., Drew, M.S., Funt, B.V.: Color constancy: generalized diagonal transforms suffice. Journal of the Optical Society of America, A 11, 3011–3019 (1994)
4. Finlayson, G.D., Hordley, S.D., Xu, R.: Convex programming colour constancy with a diagonal-offset model. In: IEEE ICIP, vol. 3, pp. 948–951 (2005)
5. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice-Hall, Englewood Cliffs (2007)
6. Healey, G., Jain, A.: Retrieving multispectral satellite images using physics-based invariant representations. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 842–848 (1996)
7. Kaneko, S., Murase, I., Igarashi, S.: Robust image registration by increment sign correlation. Pattern Recognition 35(10), 2223–2234 (2002)
8. Kaneko, S., Satoh, Y., Igarashi, S.: Using selective correlation coefficient for robust image registration. Pattern Recognition 36(5), 1165–1173 (2003)
9. Lenz, R., Tran, L.V., Meer, P.: Moment based normalization of color images. In: IEEE 3rd Workshop on multimedia signal processing, pp. 103–108 (1998)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
11. Mindru, F., Tuytelaars, T., Gool, L.V., Moons, T.: Moment invariants for recognition under changing viewpoint and illumination. Computer Vision and Image Understanding 94, 3–27 (2004)
12. Montoliu, R., Pla, F.: Generalized least squares-based parametric motion estimation. Computer Vision and Image Understanding 113(7), 790–801 (2009)
13. Shao, L., Brady, M.: Invariant salient regions based image retrieval under viewpoint and illumination variations. J. Vis. Commun. Image R. 17, 1256–1271 (2006)
14. Torr, P.H.S., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78, 138–156 (2000)
15. West, G., Brill, M.H.: Necessary and sufficient conditions for von kries chromatic adaptation to give color constancy. J. Math. Biol. 15, 249–258 (1982)
16. Wyszecki, G., Stiles, W.S.: Color science. John Wiley & Sons, Chichester (1967)
17. Xingming, Z., Huangyuan, Z.: An illumination independent eye detection algorithm. In: IEEE ICPR, vol. 1, pp. 342–345 (2006)
18. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21, 997–1000 (2003)

A Novel Approach to Robust Background Subtraction

Walter Izquierdo Guerra and Edel García-Reyes

Advanced Technologies Application Center (CENATAV), 7a ♯ 21812 e/ 218 y 222,
Rpto. Siboney, Playa, C.P. 12200, La Habana, Cuba
{wizquierdo,egarcia}@cenatav.co.cu

Abstract. Nowadays, background model does not have any robust solution and constitutes one of the main problems in surveillance systems. Researchers work in several approaches in order to get better background pixel models. This is a previous step to apply the background subtraction technique and results are not as good as people expect. We propose a novel approach to the background subtraction technique without a strong dependence of the background pixel model. We compare our algorithm versus Wallflower algorithm [1]. We use the standards deviation of the difference as an independent initial parameter to reach an adjusted threshold for every moment. This solution is more efficient computationally than the wallflower approach.

1 Introduction

Surveillance systems are interested in the problem of segmenting moving objects in video sequences. Background subtraction technique is one of the most used approaches. This algorithm compares the current image versus a background image obtained by a previous processing of the pixel history. The pixels where the difference was greater than a threshold are marked as foreground pixels. That is the main principle for this technique. In our opinion this kind of algorithms may be separate in two main steps: background maintenance and segmenting criteria.

The background maintenance is the step where the background is modeled. Next, it is predicted an expected image according to his model. In general, this is the main feature that distinguishes methods. The current models report a lot of errors to predict the background. Some researchers have produced states of the art of the existent methods in last years[1], [2],[3] and [4].

The second step (segmenting criteria) has evolved since a simple priori threshold [5] to a more complex system as [1].In general, this step is based on the first one. Some variables are inferred from the background maintenance phase in order to obtain an automatic threshold to segment foreground pixels.

One of the most popular algorithms is the Gaussian mixture model. In [6], the authors explain a detailed version of it. At present, there are authors trying to improve this method because it has a great number of advantages.For example, the authors of [7] propose an approach that combines the Gaussian mixture

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 69–76, 2009.

model with a Markov random fields smoothness. That algorithm has a great computational cost. It fixes a lot of parameters. That become the method in a scene depended method. A survey with a great amount of approaches can be found in [8]. Most of them try to solve the problem of robust background maintenance, but the number of situations that can be observed in an image sequence is colossal.

The main problems are presented in [1]. We are going to focus our work in seven of them: moved object, time of day, light switch, waving trees, camouflage, bootstrap and foreground aperture. There are works which try to solve other problems. For example [9] shows an algorithm to solve the shadows in the image.

We propose a novel method to solve the problems of background subtraction technique. Our algorithm does not have a strong dependence of background maintenance. Algorithms like Cutler and Wallflower [4] use a general threshold in order to segment the difference between the current images and the background. However, the Cutler uses a fixed value calculated offline for all frames. Wallflower is more similar to our method because its threshold is calculated dynamically, but taking into account p previous frames. In our case, we only use the standard deviation of the difference as an independent initial parameter to reach an adjusted threshold for every moment.

This paper is divided in 4 sections. Section 1 is an overview of this work. Section 2 describes our algorithm and some theoretical topics about it. Section 3 presents how we use our approach and the comparison of our results versus Wallflower algorithm. Lastly, section 4 contains the conclusions and future work.

2 Our Approach

Nowadays, the researchers are modeling at pixel level. After that, they subtract the current image from the background (”subtracted image”). Lastly, they apply a threshold(obtained in modeling phase) for each pixel and classify those pixels as foreground or background. Then, they process the obtained images at region and frame levels. At region level, they use connected component analysis, morphological and texture tools, among others. But there are not robust mathematical models at those levels.

We present a novel approach for the detection problem in video image sequences. It is based on the background subtraction. The focus is not based on a new modeling tool at pixel level. We are going to use a simple background maintenance strategy.

For predicting the value of one pixel we are going to use the mean of this pixel in the time(this is a very simple model at pixel level). For this purpose we calculate the mean per pixel of the N first frames as initial background, using an update mechanism as follows:

$$\mu_{r,c}(t+1) = \mu_{r,c}(t) * \frac{N-1}{N} + \frac{I_{r,c}(t)}{N} \tag{1}$$

Where $\mu_{r,c}(t)$ and $I_{r,c}(t)$ are the estimated mean and the pixel value, in the pixel (r, c) (row, column) in the frame t, respectively. Also, we present a very similar correction to the defined in [1]. If the average of an image decreases until exceeds certain threshold T, the algorithm calculates a second background automatically, independent of the first, keeping both in memory to use them later in two directions, when average is over T and below it. The application of this correction makes the algorithm works fine for fast illumination changes.

We need to subtract the background in order to apply our model. We start from a calculated background, using the equation (1). Next, we are going to obtain the "subtracted image" ($S_{r,c}$) as:

$$S_{r,c}(t+1) = I_{r,c}(t+1) - \mu_{r,c}(t+1) \tag{2}$$

Only two cases can be observed: the current image is a background image or there are objects in the scene.

In the first case(background image) we suppose there is only noise or illumination changes. This way, when numerical values of the resulting image("subtracted image") are plotted, we should obtain a Gaussian distribution centered in zero if only exists noise or centered in δ, if there was an illumination change, being δ the magnitude of this change.

Our hypothesis is that it is possible to define an automatic threshold which depends on the global image noise to detect the moving objects.

The three channel colors (R,G,B) of the values of $S_{r,c}$ are adjusted to a three Gaussian with parameters μ_R (mean of the channel R), μ_G (mean of the channel G), μ_B (mean of the channel B), σ_R (standard deviation of the channel R), σ_G (standard deviation of the channel G) and σ_B (standard deviation of the channel B) that characterize them.

Figure 1a shows the obtained result, adjusting the "subtracted image" from a video sequence without object presence, with the obtained background from the preceding frames. None of the obtained adjusts had a regression coefficient under 0.9. Our gaussian distribution is centered in zero because there are only noise.

Figure 1b shows an illumination change. Notice that mean is not in zero.

Then, we propose an automatic threshold $T = k * \sigma$ dependent on the global image noise level to segment the objects.

The binary image is obtained as:

$$B_{r,c}(t+1) = \begin{cases} 1 \; if |S_{r,c}(t+1) - \delta(t+1)| \geq k * \sigma(t+1) \\ \\ 0 \; if |S_{r,c}(t+1) - \delta(t+1)| < k * \sigma(t+1) \end{cases} \tag{3}$$

Where $B_{r,c}$ is the binary image obtained as a first level of detection. We calculate $\delta(t+1)$(magnitude of change) as the mean of the matrix $S_{r,c}(t+1)$. The detected objects are labeled with value 1. In other words, when a pixel is out of our interest region, we say this is a pixel which does not belong to our distribution and we label it as foreground.

After, we performance a correction in the histogram displacement by the effect of illumination changes($|S_{r,c}(t+1) - \delta(t+1)|$), this distribution is centered in zero. We may find two kind of noise's extreme behavior in the subtracted image.

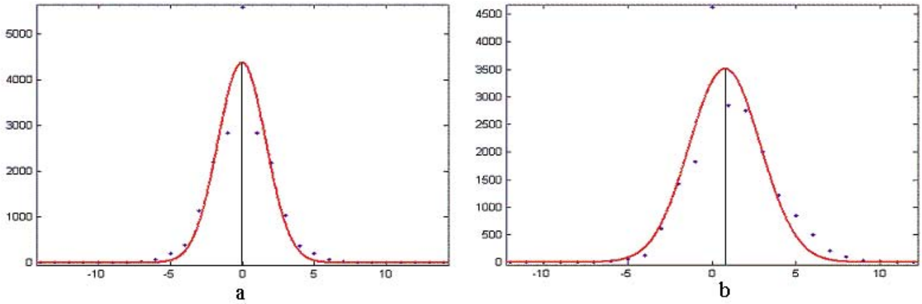


Fig. 1. A "Subtracted image" adjusted to a Gaussian distribution function centered in zero(only noise is observed). b "Subtracted image" adjusted to a Gaussian distribution function centered in δ(illumination change).

One, the current image has very low noise level which show a narrow histogram, in this case with a threshold near to zero we assure low false positive rate for the pixel label as background. Second, the current image has very high noise level and with a threshold more distant to zero we assure low false positive rate for the pixel label as foreground. In other words, if we fix k, when the current image has a lot of noise we obtain a T value $(k * \sigma)$ so big that all pixels will belong to our gaussian distribution model and they will be classified as background pixels and when the current image is free of noise our threshold will tend to zero, then most of the pixels will be out of the interest region and will be classified as foreground.

Then, we need an adaptive threshold in order to deal with different noise levels. The following step will try to apply a regional analysis to solve the confusion re-labeling the pixels whose values are higher than the threshold in the first case and the pixels lower than the threshold in the second one.

Then, we defined k dependent on σ.

$$k = \exp(-(\frac{\sigma(t+1) - 0.23 * n}{0.12 * n})^2) + 3 * \exp(-(\frac{\sigma(t+1)}{0.08 * n})^2) \qquad (4)$$

Where n is the color scale(256 in our case).

This is a semi-empiric equation constructed in order to solve the problem explained before. Notice that if σ is a big value, k is going to be a small value to smooth the noise effect.

If σ value is small then k is going to be big(for example $k > 3$) and we can ensure, according to our gaussian model(see figure 2), that more than 99,7% of pixels belongs to our model are inside our interest region and all pixels do not belong to our distribution are foreground pixels.

In the second case, if σ value is big then k is going to tend to zero. This reduces a lot our interest region and we can ensure that all pixels belong to our distribution are background pixels.

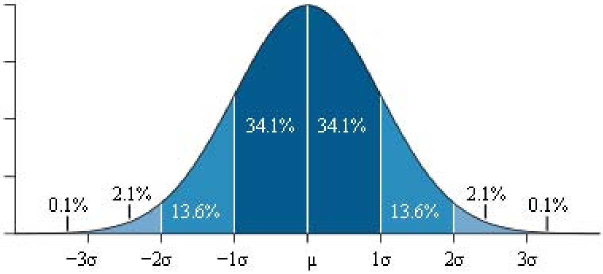


Fig. 2. Dark blue is less than one standard deviation from the mean. For the normal distribution, this accounts for about 68% of the set (dark blue) while two standard deviations from the mean (medium and dark blue) account for about 95% and three standard deviations (light, medium, and dark blue) account for about 99.7%. Extracted from [10].

We need to relabel the pixels, at region level, in order to recover foreground pixels that belong to gaussian distribution (in the first case) and the background pixels do not belong(in the second case).

The processing continues convolving the image with a filter of 3x3 to remove isolated pixels. Next, we apply a connected components algorithm. We keep the regions greater or equal than a size estimated for a person (suitable for each scene) and with more than 30 % of foreground pixels inside the minor convex polygon that surround the component. The binary image obtained from this step is designated as $P_{r,c}$.

In order to relabel, we are going to join the connected components with certain degree of similarity. For this purpose, we define a criteria to relabel the pixels taking into account the mean color of the object and the spatial distance of the pixel to the object.

$$M_{r,c}(t+1) = E_{r,c}(t+1) + R * C_{r,c}(t+1) \tag{5}$$

$$C_{r,c}(t+1) = |I_{r,c}(t+1) - m(t+1)| \tag{6}$$

$$m(t+1) = \frac{1}{N_0(t+1)} \sum_{r,c} P_{r,c}(t+1) * I_{r,c}(t+1) \tag{7}$$

Where N_0 is the number of pixels distinct of zero in the binary image $P_{r,c}(t+1)$ and R is a constant that depends of the pedestrian dimension in the surveillance scene. Thus, m is the mean of pixel values labeled as object, in the binary image P, in the current image. $E_{r,c}$ is the minor distance from pixel (r, c) to a labeled pixel as object(distance transform [11]). We threshold to obtain the binary image A as:

$$A_{r,c}(t+1) = \begin{cases} 1\ if|M_{r,c}(t+1)| \leq R \\ \\ 0\ if|M_{r,c}(t+1)| > R \end{cases} \tag{8}$$

Where A is the resulting matrix of our algorithm will return. As we can observe in equations (5), (6) and (8) if $C_{r,c}(t+1) > 1$ then $A_{r,c}(t+1) = 0$. This is because our criteria is very susceptible to color change.

3 Experimental Analysis

In order to apply our algorithm to the wallflower's dataset [1], we adjust the constant parameter R. As was mentioned above, this parameter depends on the dimensions of the object we want to detect. We use two values of R in this dataset because there are two different dimensions of the human body: the first one, presented in the Waving Trees, Camouflage and Foreground Aperture video sequences, which is bigger than the dimension presented in the other video sequences.

With this correction, we apply our method. The results are shown in Table1. Here we have the wallflower algorithm results [1] and ours over the wallflower dataset.

In the figure 3, we compare our results versus wallflower's results. Wallflower's algorithm is a very famous method and its dataset is one of the most used to compare algorithms. The first row of pictures is hand-segmented images. Look at the pictures and notice that our algorithm works much better than wallflower's method.

As we can observe in Table 1, our algorithm work better than wallflower algorithm. It reduces to 56 % the total of errors they reported.

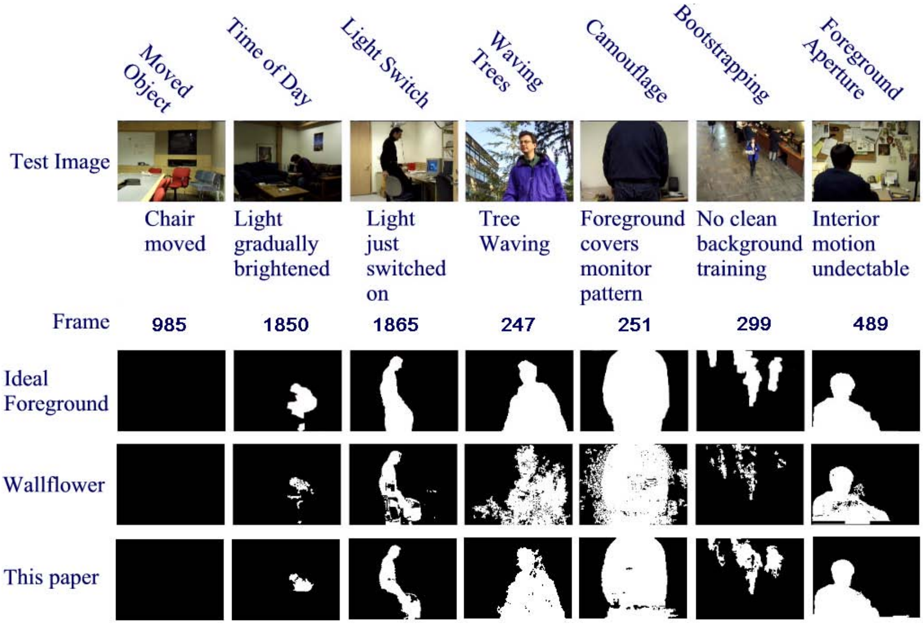


Fig. 3. Visual results from Wallflower and this paper

Table 1. Comparison of Wallflower and this paper

Algorithm	Error Type	moved object	time of day	light switch	waving trees	camouflage	bootstrap	foreground aperture	Total Errors
Wallflower	false neg.	0	961	947	877	229	2025	320	11448
	false pos.	0	25	345	1999	2706	365	649	
This paper	false neg.	0	1030	1308	164	518	907	236	5906
	false pos.	0	3	385	333	384	565	73	

As we show in Table 1, in light switch image, there are a great amount of false negative pixels. In our opinion, this does not constitute an error of our algorithm because most of them are pixels belonged to a chair, that wallflower dataset report as an object. We consider that the chair is background in the scene.

4 Conclusions

In this paper, we present a novel approach to the background subtraction technique. The global threshold used to segmenting the moving objects is dependent on the current image noise level and it is automatically calculated applying an empirical formula. We need to set only one parameter (R) for our algorithm. We experimentally compare our approach against the wallflower algorithm and we obtained better results, as showed visually in figure 3, and numerically in table 1. Our future research direction is to combine our algorithm with a most robust tool to model the pixel history.

References

1. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Seventh International Conference on Computer Vision, vol. 1, p. 255 (1999)
2. Sen, Kamath, C.: Robust techniques for background subtraction in urban traffic video, vol. 5308, pp. 881–892. SPIE, San Jose (2004)
3. Ribeiro, H.N., Hall, D., Nascimento, J., Ribeiro, P., Andrade, E., Moreno, P., Pesnel, S., List, T., Emonet, R., Fisher, R.B., Santos Victor, J., Crowley, J.L.: Comparison of target detection algorithms using adaptive background models. In: Proc. 2nd Joint IEEE Int. Workshop on Visual Surveillance and VisLab-TR 13/2005, pp. 113–120 (2005)
4. Mcivor, A.M.: Background subtraction techniques. In: Proc. of Image and Vision Computing, pp. 147–153 (2000)
5. Heikkilä, J., Silvén, O.: A real-time system for monitoring of cyclists and pedestrians. Image and Vision Computing 22(7), 563–570 (2004)
6. Power, P.W., Schoonees, J.A.: Understanding background mixture models for foreground segmentation. In: Proceedings Image and Vision Computing New Zealand, p. 267 (2002)

7. Schindler, K., Wang, H.: Smooth foreground-background segmentation for video processing. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 581–590. Springer, Heidelberg (2006)
8. Bouwmans, T., Baf, F.E., Vachon, B.: Background modeling using mixture of gaussians for foreground detection - a survey. Recent Patents on Computer Science 1, 219–237 (2008)
9. Kaewtrakulpong, P., Bowden, R.: An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS 2001, Video Based Surveillance Systems: Computer Vision and Distributed Processing (2001)
10. Wikipedia, http://en.wikipedia.org/wiki/normaldistribution
11. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB. Prentice-Hall, Inc., Upper Saddle River (2003)

Automatic Choice of the Number of Nearest Neighbors in Locally Linear Embedding

Juliana Valencia-Aguirre, Andrés Álvarez-Mesa, Genaro Daza-Santacoloma, and Germán Castellanos-Domínguez

Control and Digital Signal Processing Group, Universidad Nacional de Colombia, Manizales, Colombia
{jvalenciaag,amalvarezme,gdazas,cgcastellanosd}@unal.edu.co

Abstract. Locally linear embedding (LLE) is a method for nonlinear dimensionality reduction, which calculates a low dimensional embedding with the property that nearby points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space [1]. LLE algorithm needs to set up a free parameter, the number of nearest neighbors k. This parameter has a strong influence in the transformation. In this paper is proposed a cost function that quantifies the quality of the embedding results and computes an appropriate k. Quality measure is tested on artificial and real-world data sets, which allow us to visually confirm whether the embedding was correctly calculated.

1 Introduction

In many pattern recognition problems the characterization stage generates a big amount of data. There are several important reasons for reducing the feature space dimensionality, such as, improve the classification performance, diminish irrelevant or redundancy information, find out underlying data structures, obtain a graphical data representation for visual analysis, etc [2]. Several techniques for dimensionality reduction have been developed, traditionally these techniques are linear [3] and they can not correctly discover underlying structures of data lie on nonlinear manifolds. In order to solve this trouble a nonlinear mapping method called locally linear embedding (LLE) was proposed in [4,5].

This method requires to manually set up three free parameters, the dimensionality of embedding space m, the regularization parameter α, and the number of nearest neighbors k for local analysis [6]. There are two previous approaches for choosing k. Kouropteva et. al [7] presented a hierarchical method for automatic selection of an optimal parameter value based on the minimization of the residual variance. Nonetheless, the residual variance can not quantify the local geometric structure of data. Besides, Goldberg and Ritov [8] display a novel measure based on Procrustes rotation that enables quantitative comparison of the output of manifold-based embedding algorithms, the measure also serves as a natural tool for choosing dimension-reduction parameters. The local procrustes

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 77–84, 2009.

measure preserves local geometric structure but does not consider the global behavior of the manifold.

In this paper is proposed an automatic method for choosing the number of nearest neighbors, which is done by means of computing a cost function that quantifies the quality of embedding space. This function takes into account local and global geometry preservation. Proposed approach is experimentally verified on 2 artificial data sets and 1 real-world data set. Artificial data sets allow to visually confirm whether the embedding was correctly calculated, and real-world data set was used for visualization of multidimensional samples.

2 Locally Linear Embedding

Let $\mathbf{X}$ the input data $p \times n$ matrix, where the sample vectors $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$ are had. Data live on or close to a non-linear manifold and that is well-sampled. Besides, each point and its neighbors lie on a locally linear patch. LLE algorithm has 3 steps. First, search the k nearest neighbors per point, as measured by Euclidean distance. If k is set too small, the mapping will not reflect any global properties; if it is too high, the mapping will lose its nonlinear character and behave like traditional PCA [6]. Second, each point is represented as a weighted linear combination of its neighbors [9], that is, we calculate weights $\mathbf{W}$ that minimize reconstruction error

$$\varepsilon\left(\mathbf{W}\right) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j=1}^{n} w_{ij}\mathbf{x}_j\|^2, \tag{1}$$

subject to an sparseness constraint, $w_{ij} = 0$ if $\mathbf{x}_j$ is not $k-$neighbor of $\mathbf{x}_i$, and an invariance constraint $\sum_{j=1}^{n} w_{ij} = 1$. For a particular data point $\mathbf{x} \in \mathbb{R}^p$ and its k nearest neighbors $\boldsymbol{\eta}$. Let $\mathbf{G}$ the Gram $k \times k$ matrix, where its elements are $G_{jl} = \left\langle \left(\mathbf{x} - \boldsymbol{\eta}_j\right), \left(\mathbf{x} - \boldsymbol{\eta}_l\right) \right\rangle,\ j = 1, \ldots, k;\ l = 1, \ldots, k$. Then rewriting (1)

$$\varepsilon = \mathbf{w}^\top \mathbf{G} \mathbf{w} \quad \text{s.t.} \quad \sum\nolimits_{j=1}^{n} w_j = 1. \tag{2}$$

The solution of (2) is obtained by solving an eigenvalue problem. Employing Lagrange theorem $\mathbf{w} = (1/2)\,\lambda \mathbf{G}^{-1}\mathbf{1}$, where $\lambda = 2/\left(\mathbf{1}^\top \mathbf{G}^{-1}\mathbf{1}\right)$. When Gram matrix $\mathbf{G}$ is singular (or close), the result of least squares problem for finding $\mathbf{w}$ does not have unique solution. So, it is necessary to regularize $\mathbf{G}$ before finding $\mathbf{w}$. In [1,5] is proposed to calculate the regularization of $\mathbf{G}$ as $G_{jl} \leftarrow G_{jl} + \alpha$ where $\alpha = \delta_{jl}\left(\Delta^2/k\right)\operatorname{tr}\left(\mathbf{G}\right)$, being $\delta_{jl} = 1$ if $j = l$ and 0 in other case, $\Delta^2 \ll 1$. However, Δ must be empirically tuned, in [1] is advised to employ $\Delta = 0.1$.

In third step low dimensional embedding is calculated. Using $\mathbf{W}$, the low dimensional output $\mathbf{Y}$ is found by minimizing (3)

$$\Phi\left(\mathbf{Y}\right) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j=1}^{n} w_{ij}\mathbf{y}_j\|^2, \tag{3}$$

subject to $\sum_{i=1}^{n} \mathbf{y}_i = \mathbf{0}$ and $\sum_{i=1}^{n} \mathbf{y}_i \mathbf{y}_i^\top / n = \mathbf{I}_{m\times m}$, where $\mathbf{Y}$ is the embedding data $n \times m$ matrix (being $m \leq p$), and $\mathbf{y}_i \in \mathbb{R}^m$ is the output sample vector.

Let $\mathbf{M} = \left(\mathbf{I}_{n\times n} - \mathbf{W}^\top\right)\left(\mathbf{I}_{n\times n} - \mathbf{W}\right)$, and rewriting (3) to find $\mathbf{Y}$,

$$\Phi\left(\mathbf{Y}\right) = \mathrm{tr}\left(\mathbf{Y}^\top \mathbf{M} \mathbf{Y}\right) \quad \text{s.t.} \quad \begin{cases} \mathbf{1}_{1\times n}\mathbf{Y} = \mathbf{0}_{1\times n} \\ \frac{1}{n}\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_{m\times m} \end{cases} \tag{4}$$

it is possible to calculate $m+1$ eigenvectors of $\mathbf{M}$, which are associated to $m+1$ smallest eigenvalues. First eigenvector is the unit vector with all equal components, which is discarded. The remaining m eigenvectors constitute the m embedding coordinates found by LLE.

3 Measure of Embedding Quality

When dimensionality reduction technique is computed is necessary to establish a criteria for knowing if its results are adequate. In LLE is searched a transformation that preserves the local data geometry and global manifold properties.

The quality of an output embedding could be judged based on a comparison to the structure of the original manifold. However, in the general case, the manifold structure is not given, and it is difficult to estimate accurately. As such ideal measures of quality cannot be used in the general case, an alternate quantitative measure is required [8].

In [7], the residual variance is employed as a quantitative measure of the embedding results. It is defined as

$$\sigma_R^2(D_\mathbf{X}, D_\mathbf{Y}) = 1 - \rho^2_{D_\mathbf{X} D_\mathbf{Y}}, \tag{5}$$

where ρ^2 is the standard linear correlation coefficient, taken over all entries of $D_\mathbf{X}$ and $D_\mathbf{Y}$; $D_\mathbf{X}$ and $D_\mathbf{Y}$ are the matrices for Euclidean distances in $\mathbf{X}$ and $\mathbf{Y}$, respectively. $D_\mathbf{Y}$ depends on the number of neighbors selected k. According to [7], the lowest residual variance corresponds to the best high-dimensional data representation in the embedded space. Hence, the number of neighbors can be computes as

$$k_{\sigma_R^2} = \arg\min_k(\sigma_R^2(D_\mathbf{X}, D_\mathbf{Y})). \tag{6}$$

On the other hand, in [8] is proposed to compare a neighborhood on the manifold and its embedding using the Procrustes statistic as a measure for qualifying the transformation. This measures the distance between two configurations of points and is defined as $P\left(\mathbf{X},\mathbf{Y}\right) = \sum_{i=1}^{n} \left\|\mathbf{x}_i - \mathbf{A}\mathbf{y}_i - \mathbf{b}\right\|^2$, being $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$ and $\mathbf{b} \in \mathbb{R}^m$. The rotation matrix $\mathbf{A}$ can be computed from $\mathbf{Z} = \mathbf{X}^\top\mathbf{H}\mathbf{Y}$, where $\mathbf{H} = \mathbf{I} - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$, $\mathbf{1}$ is a $n \times 1$ column vector, and $\mathbf{H}$ is the centering matrix. Let $\mathbf{U}\mathbf{L}\mathbf{V}^\top$ be the singular-value decomposition of $\mathbf{Z}$, then $\mathbf{A} = \mathbf{U}\mathbf{V}^\top$, the translation vector $\mathbf{b} = \overline{\mathbf{x}} - \mathbf{A}\overline{\mathbf{y}}$, where $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are the sample means of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Let $\left\|\cdot\right\|_F$ the Frobenius norm, so $P\left(\mathbf{X},\mathbf{Y}\right) = \|\mathbf{H}(\mathbf{X} - \mathbf{Y}\mathbf{A}^\top)\|_F^2$.

In order to define how well an embedding preserves the local neighborhoods using the Procrustes statistic $P_L(\mathbf{X}_i, \mathbf{Y}_i)$ of each neighborhood-embedding pair

$(\mathbf{X}_i, \mathbf{Y}_i)$. $P_L(\mathbf{X}_i, \mathbf{Y}_i)$ estimates the relation between the entire input neighborhood and its embedding as one entity, instead of comparing angles and distances within the neighborhood with those within its embedding. A global embedding that preserves the local structure can be found by minimizing the sum of the Procrustes statistics of all neighborhood-embedding pairs [8], taking into account an scaling normalization, which solves the problem of increased weighting for larger neighborhoods, so

$$R_N(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} P_L(\mathbf{X}_i, \mathbf{Y}_i) / \left\| \mathbf{H_L} \mathbf{X}_i \right\|_F^2, \tag{7}$$

where $\mathbf{H_L} = \mathbf{I} - \frac{1}{k} \mathbf{1}\mathbf{1}^\top$; $\mathbf{1}$ is a $k \times 1$ vector. Therefore, the number of nearest neighbors can be calculated as

$$k_{R_N} = \arg\min_k (R_N(\mathbf{X}, \mathbf{Y})). \tag{8}$$

In this work, we propose an alternative measure for quantifying the embedding quality. This measure attempts to preserve the local geometry and the neighborhood co-location, identifying possible overlaps on the low dimensional space. And it is defined as

$$C_I(\mathbf{X}, \mathbf{Y}) = \\ \frac{1}{2n} \sum_{i=1}^{n} \left\{ \frac{1}{k} \sum_{j=1}^{k} \left(D_{(\mathbf{x}_i, \boldsymbol{\eta}_j)} - D_{(\mathbf{y}_i, \boldsymbol{\phi}_j)} \right)^2 + \frac{1}{k_n} \sum_{j=1}^{k_n} \left(D_{(\mathbf{x}_i, \boldsymbol{\theta}_j)} - D_{(\mathbf{y}_i, \boldsymbol{\gamma}_j)} \right)^2 \right\}, \tag{9}$$

where D is an standardized Euclidean distance to obtain a maximum value equal to one. For example $D_{(\mathbf{x}_i, \boldsymbol{\eta}_j)}$ is the distance calculated between the observation $\mathbf{x}_i$ and each one of its k neighbors on the input space.

Once the embedding, for each point $\mathbf{y}_i \in \mathbb{R}^m$ a set $\boldsymbol{\beta}$ of k nearest neighbors is calculated, and the projection $\boldsymbol{\phi}$ of $\boldsymbol{\eta}$ is found. The neighbors computed in $\boldsymbol{\beta}$ that are not neighbors in $\boldsymbol{\eta}$ conform a new set $\boldsymbol{\gamma}$, that is $\boldsymbol{\gamma} = \boldsymbol{\beta} - (\boldsymbol{\beta} \cap \boldsymbol{\phi})$. The size of $\boldsymbol{\gamma}$ is k_n. Besides, the projections of the elements of $\boldsymbol{\gamma}$ in $\mathbf{X}$ conform the set $\boldsymbol{\theta}$ of k_n neighbors. In an ideal embedding $C_I(\cdot) = 0$.

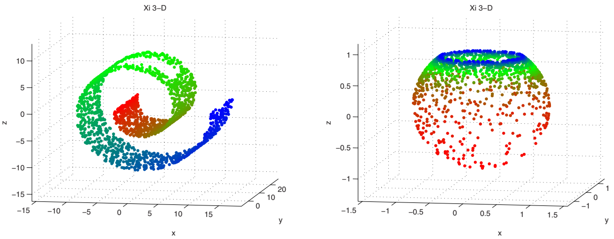
The first term in (9) quantifies the local geometry preservation. The k nearest neighbors of $\mathbf{x}_i$ chosen on the high dimensional space $\mathbf{X}$ are compared against their representations in the embedded space $\mathbf{Y}$. The second term computes the error produced by possible overlaps in the embedded results, which frequently occurs when the number of neighbors is strongly increased, and global properties of the manifold are lost. The number of nearest neighbors can be found as

$$k_{C_I} = \arg\min_k (C_I(\mathbf{X}, \mathbf{Y})). \tag{10}$$

4 Experimental Background

4.1 Tests on Artificial Data Sets

Two different manifold are tested, which allow to visually confirm whether the embedding was correctly calculated. The Swiss Roll with Hole data set with



(a) Swiss Roll with Hole (b) Fishbowl

Fig. 1. Artificial Data Sets

2000 samples (Fig. 1(a)) and the Fishbowl data set with uniform distribution in embedding space and 1500 samples (Fig. 1(b)).

In order to quantify the embedding quality and find the number of nearest neighbors needed for a faithful embedding, LLE is computed by varying k in the subset $k \in \{3, 4, 5, \ldots, 250\}$, fitting the dimensionality of the embedded space to $m = 2$. The embedding quality is computed according to (5), (7) and (9). The number of nearest neighbors is found by means of (6), (8) and (10).

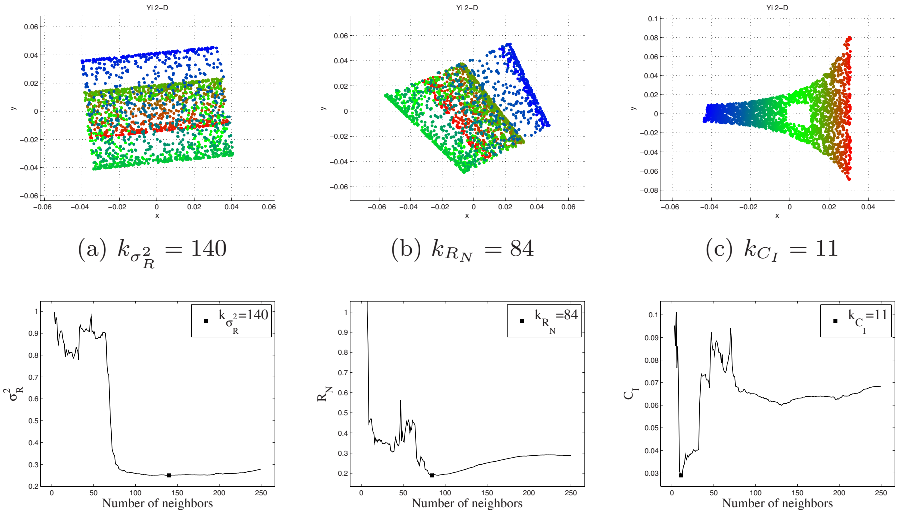
In Figures 2(a), 2(b), and 2(c) are presented the embedding results for the Swiss Roll with Hole data set using an specific number of neighbors in accordance with each one of the criteria above pointed out. Similarly, in Figures 3(a), 3(b), and 3(c) the embedding results for the Fishbowl data set are displayed. For these artificial data sets only our approach (10) finds appropriate embedding results preserving local and global structure. In the case of the Swiss Roll with Hole, the criteria (6) and (8) produce overlapped embeddings. Besides, in Fishbowl, the unfolding results obtained by means of (6) and (8) are wrong, those are similar to PCA and then local structure is lost.

On the other hand, Figures 2(d), 2(e), 2(f), and 3(d), 3(e), 3(f), show curves of the cost function value versus the number of neighbors. The full fill square on the curves is the global minimum of the function and corresponds to the number of nearest neighbors chosen for the embedding.

4.2 Tests on Real-World Data Sets

We use Maneki Neko pictures, which is in the Columbia Object Image Library (COIL-100) [10]. There are 72 RGB-color images for this object in PNG format. Pictures are taken while the object is rotated 360 degrees in intervals of 5 degrees. The image size is 128 × 128. In Fig. 4 are shown some examples. We transform these color images to gray scale, next the images were subsampled to 64 × 64 pixels. Then, we have input space of dimension $p = 8192$ and $n = 72$.

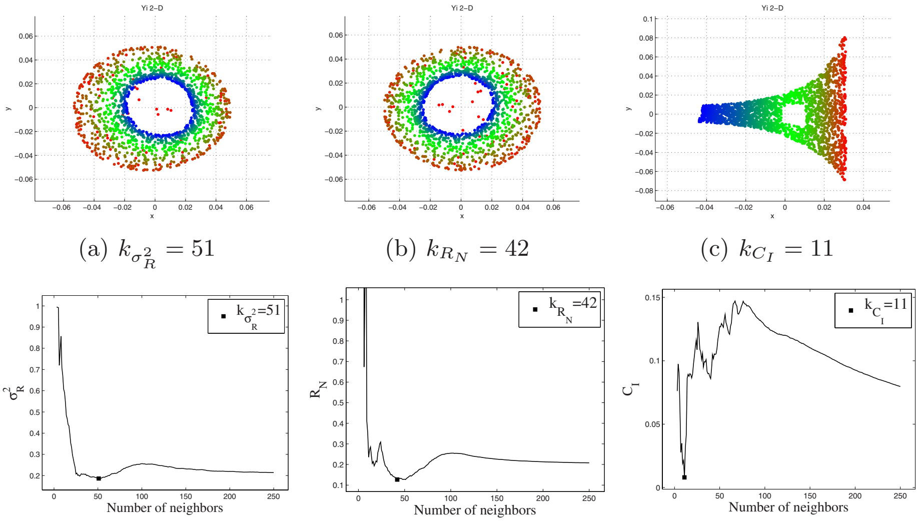
In order to quantify the embedding quality and find the number of neighbors needed for a faithful embedding, LLE is computed by varying k in the subset $k \in \{3, 4, 5, \ldots, 36\}$, fitting the dimensionality of the embedded space to $m = 2$. In Fig. 5 are shown the embedding results for the Maneki Neko data set and



(a) $k_{\sigma^2_R} = 140$ (b) $k_{R_N} = 84$ (c) $k_{C_I} = 11$

(d) $\sigma^2_R\,(k)$ (e) $R_N\,(k)$ (f) $C_I\,(k)$

Fig. 2. Results for the Swiss Roll with Hole Data Set

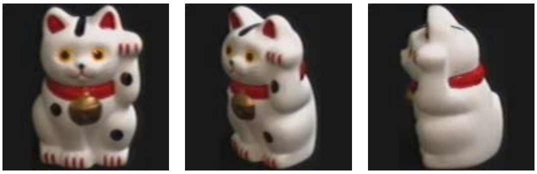


(a) $k_{\sigma^2_R} = 51$ (b) $k_{R_N} = 42$ (c) $k_{C_I} = 11$

(d) $\sigma^2_R\,(k)$ (e) $R_N\,(k)$ (f) $C_I\,(k)$

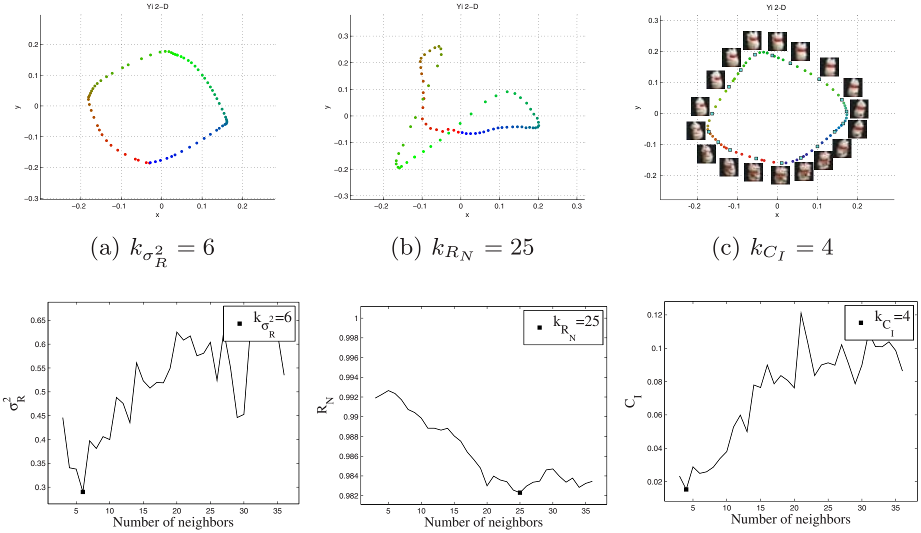
Fig. 3. Results for the Fishbowl Data Set

its corresponding cost function curves, which allow to establish the number of neighbors employed in the transformation, according to (6), (8) and (10).



(a) 0° (b) 45° (c) 90°

Fig. 4. Examples from Maneki Neko Data Set



(a) $k_{\sigma_R^2} = 6$ (b) $k_{R_N} = 25$ (c) $k_{C_I} = 4$

(d) $\sigma_R^2\,(k)$ (e) $R_N\,(k)$ (f) $C_I\,(k)$

Fig. 5. Results of Maneki Neko Data Set

5 Discussion

From the obtained results on artificial data sets (Fig. 2(d) and 3(d)) employing the cost function (6) proposed in [7], as a global trend, it is possible to notice if the size of the neighborhood is augmented, the value of σ_R^2 is diminished. Because a transformation employing a large number of neighbors results in a linear transformation and the residual variance does not quantify the local geometric structure, then this measure can not identify a suitable number of neighbors k. Figures 2(a) and 3(a) show some examples of this situation. In this case, data in low-dimensional space are overlapped and the cost function (6) does not detect it. Nevertheless Fig. 5(d) does not display the trend above pointed out and allows to calculate an appropriate embedding (Fig. 5(a)). The inconsistent results obtained by using the residual variance make of it an unreliable measure. In [7] the ambiguous results are attributed to the fact that Euclidean distance becomes an unreliable indicator for proximity.

On the other hand, the measure proposed in [8] takes into account the local geometric structure but does not consider the global behavior of the manifold.

Then, far neighborhoods can be overlapped in the low-dimensional space and this measure shall not detect it, which can be seen in Fig. 2(b), 3(b), and 5(b). The measure of embedding quality presented here (10) computes a suitable number of neighbors for both artificial and real-world manifolds. Besides, the embedding results calculated using this criterion are in accordance to the expected visual unfolding (Fig 2(c), 3(c), and 5(c)). The proposed measure preserves the local geometry of data and the global behavior of the manifold.

6 Conclusion

In this paper a new measure for quantifying the embedding quality using LLE is proposed. This measure is employed as a criterion for choosing automatically the number of nearest neighbors needed for the transformation. We compare the new cost function against two methods presented in the literature. The best embedding results (visually confirmed) were obtained using the approach exposed, because it preserves the local geometry of data and the global behavior of the manifold.

Acknowledgments. This research was carried out under grants provided by the project *Representación y discriminación de datos funcionales empleando inmersiones localmente lineales* funded by Universidad Nacional de Colombia, and a Ph. D. scholarship funded by Colciencias.

References

1. Saul, L.K., Roweis, S.T.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Machine Learning Research 4 (2003)
2. Carreira-Perpiñan, M.A.: A review of dimension reduction techniques (1997)
3. Webb, A.R.: Statistical Pattern Recognition, 2nd edn. Wiley, USA (2002)
4. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
5. Saul, L.K., Roweis, S.T.: An introduction to locally linear embedding. AT&T Labs and Gatsby Computational Neuroscience Unit, Tech. Rep. (2000)
6. de Ridder, D., Duin, R.P.W.: Locally linear embedding for classification. Delft University of Technology, The Netherlands, Tech. Rep. (2002)
7. Kouropteva, O., Okun, O., Pietikäinen, M.: Selection of the optimal parameter value for the locally linear embedding algorithm. In: The 1st International Conference on Fuzzy Systems and Knowledge Discovery (2002)
8. Goldberg, Y., Ritov, Y.: Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. Machine learning (2009)
9. Polito, M., Perona, P.: Grouping and dimensionality reduction by locally linear embedding. In: NIPS (2001)
10. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library: Coil-100. Columbia University, New York, Tech. Rep. (1996)

K-Medoids-Based Random Biometric Pattern for Cryptographic Key Generation

H.A. Garcia-Baleon, V. Alarcon-Aquino, and O. Starostenko

Deparment of Computing, Electronics, and Mecatronics,
Universidad de las Americas Puebla
72820 Cholula, Puebla, Mexico
{hectora.garciabn,vicente.alarcon}@udlap.mx

Abstract. In this paper we report an approach for cryptographic key generation based on keystroke dynamics and the k-medoids algorithm. The stages that comprise the approach are training-enrollment and user verification. The proposed approach is able to verify the identity of individuals off-line avoiding the use of a centralized database. The performance of the proposed approach is assessed using 20 samples of keystroke dynamics from 20 different users. Simulation results show a false acceptance rate (FAR) of 5.26% and a false rejection rate (FRR) of 10%. The cryptographic key released by the proposed approach may be used in several encryption algorithms.

Keywords: keystroke dynamics, biometrics, cryptography, k-medoids.

1 Introduction

The combination of biometrics and cryptography has attracted the attention of some researches due to the fact that this combination can bring together the better of the two worlds. The idea of combining biometrics and cryptography is not new; however, the concept is poorly developed because several biometric cryptosystems require maintaining the biometric information in a centralized database. This fact has a serious impact in the social acceptance of the biometric cryptosystems. The first practical system that integrates the iris biometrics into cryptographic applications is reported in [3]. A system that works using fingerprint identification based on a token is presented in [1]. A successful combination of face biometric and cryptography for key generation is also reported in [2]. Another research that uses on-line handwritten signatures to generate cryptographic keys is reported in [4]. Other approaches have also been reported in [8,9]. However, these approaches have also reported a poor FAR and FRR. Both performance metrics are crucial in determining if the combined system can be implemented in real scenarios.

Keystroke dynamics can be defined as the timing data that describes when a key is pressed and when a key is released as a user types at the keyboard. The recorded timing data can be processed through an algorithm to determine a primary timing pattern (PTP) for future verification. The PTP is used to verify

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 85–94, 2009.

the identity of the individual. The design of the proposed approach considers three security factors, namely, a user password, a behavioral biometric sample, and a token. It works using a 3D random distribution of the biometric data that assures also the randomness of the cryptographic key released. The 3D pattern is extracted from the 3D random biometric pattern using the k-medoids algorithm tested for different types of distances that measure similarity, namely, Manhattan, Euclidean, Chebyshev and Markowski distance. The rest of the paper is organized as follows. In Section 2, the k-medoids algorithm is described. Section 3 presents the Minkowski distance for measuring similarity. Section 4 presents the keystroke dynamics and shows how the PTP is extracted to work with the proposed approach. In Section 5, the design of the proposed approach is explained, whereas in Section 6 simulation results are reported. Finally, conclusions are reported in Section 7.

2 K-Medoids Algorithm

The k-medoids algorithm is a clustering algorithm based on the k-means algorithm and the medoidshift algorithm. Both, k-means and k-medoids, algorithms break the dataset up into k clusters [5, 6]. Also, these algorithms attempt to minimize squared error. The squared error can be defined as the distance between points labeled to be in a cluster and a point designated as the center of that cluster. The k-medoids algorithm chooses datapoints as centers instead of computing the centers as the k-means algorithm does. The k-medoids algorithm is a partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori. The k-medoids algorithm is more robust to outliers and noise compared to the k-means algorithm [6]. A medoid is defined as that object of a cluster whose average dissimilarity to the rest of the objects in that cluster is minimal. The partitioning around medoids (PAM) algorithm describes a common realization of the k-medoid clustering algorithm. The PAM algorithm is as follows:

1. *Arbitrary selection of k objects as medoid points out of n datapoints ($n > k$).*
2. *Associate each data object in the given data set to the most similar medoid to form clusters. The similarity in this step can be computed using distance measure. The distance measure used can be Euclidean, Manhattan, Chebyshev, or Minkowski distance.*
3. *Randomly select a non-medoid object named R' for each cluster.*
4. *Compute the total cost S of swapping initial medoid object to R'.*
5. *If $S < 0$, then swap initial medoid with the new one. Otherwise, the initial medoid remains.*
6. *Repeat steps 2 to 5 until there is no change in the medoids.*

The PAM algorithm is based on an iterative optimization process that evaluates the effect of swapping between the initial medoid object and the non-medoid object randomly selected. The principle of the PAM algorithm resides in step 5. It can be seen that it may require trying all objects that are currently not medoids.

Thus it represents an expensive computational cost, $Cost(k(n-k)^2)$, in each iteration. The PAM algorithm results in high quality clusters, as it may try every possible combination, working effectively for small datasets. However, due to its computational complexity, it is not practical for clustering large datasets [5,6].

3 Minkowski Distance

Formally, a similarity function aims at comparing two entities of a domain M based on their common characteristics. Similarity can be measured in several ways depending on the scale of measurement or data type. Based on the vector representation the similarity can be calculated using the concept of distance. In this paper, we use the Minkowski distance to do so. The selection of the Minkowski distance is due to the fact that it is easy to implement in software and hardware, its computational cost is lower compared with more complex distances as Mahalanobis distance, and it fits better with the characteristics of the proposed approach considering the type of data used. In general, the distance d_{ij} between any two points, $P = (x_1, x_2, ...x_n)$ and $Q = (y_1, y_2, ...y_n) \in \Re^n$, in n-dimensional space may be calculated by the equation given by Minkowski as follows [7]:

$$d_{ij} = \left(\sum_{i=1}^{n} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} \tag{1}$$

with k being the index of the coordinates, and p determining the type of distance. There are three special cases of the Minkowski distance:

- $p = 1$: this distance measure is often called city block distance, or *Manhattan distance*.
- $p = 2$: with p equalling 2 the Minkowski distance is reduced to the well-known *Euclidean distance*.
- $p = \infty$: with p equalling ∞ the Minkowski distance is reduced to the *Chebyshev distance*. In the limiting case of p reaching infinity, the resultant equation is as follows:

$$d_{ij} = \lim_{p\to\infty} \left(\sum_{i=1}^{n} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}} = max|x_{ik} - x_{jk}|_{i=1}^{n} \tag{2}$$

4 A Behavioral Biometric: Keystroke Dynamics

Keystroke dynamics is defined as the timing data that describes when a key is pressed or released as the user types at the keyboard. This behavioral biometric uses the manner and the rhythm in which a user types characters. The keystroke rhythms of a user are measured to develop a unique biometric pattern of the users typing for future verification. The recorded timing data can be processed through an algorithm to determine a PTP for future verification. The PTP is used to verify or even try to determine the identity of the individual who is

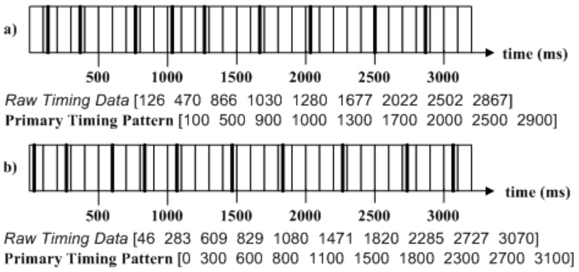


Fig. 1. Acquisition stage, a) Key pressing events pattern, b) Key releasing events pattern

producing those keystrokes. This is often possible because some characteristics of keystroke production are as individual as handwriting or a signature.

The technique used to extract the PTP used in this paper considers partitioning the acquisition time in time slots. The size of the time slot affects directly the FAR and FRR metrics. Several experiments performed showed that a size of $100ms$ for the time slot is good enough to minimize the FAR and FRR metrics as it is shown in Section 6. Figure 1 shows the timing data of an individual. In the top part, the timing data from the key pressing events is shown. The bottom part shows the timing data from the key releasing events. As can be seen, the key pressing process produces 9 events represented by the bold lines. The key releasing process produces 10 events also represented by the bold lines. It is important to notice that the first key pressed launches the acquisition stage and also the timer. It is assumed that the first key pressed event is located at zero in the time scale and thus the event is not considered in the computing of the PTP. The rest of events are located in the time scale according to the value that the timer has when the events take place. Figure 1 also depicts that the events can occur at any time within a determined time slot however the time value is rounded to the closest time slot value given in ms. This fact assures that the extracted pattern only comprises a combination of the possible discrete time values otherwise the possible time values that the event could take are infinite.

5 Proposed Approach

The successful recovering of the random cryptographic key depends on a correct combination of the user password, the behavioral biometric sample and the token, which stores the user password hash, the encrypted random distribution vectors (RDVs) used to reconstruct the 3D random biometric pattern, and the 3D pattern hash. The design presented here ensures that compromising two factors at most will not let to the attacker reveal the random biometric key.

The proposed approach is divided in two stages, namely, *training-enrollment* and *user verification*. Figure 2 shows a detailed representation of the approach. The first stage is executed when an individual is going to be enrolled for the first time to the biometric cryptosystem. This stage produces through a simple training process the information needed to verify the user in the second stage. The training process uses the keystroke dynamics biometric information obtained from the user at his enrollment. The first stage can also be executed each time that the random biometric key needs to be revoked or renewed for any security concern. The second stage, user verification, is executed each time that the user needs to be identified before the biometric cryptosystem. The training-enrollment stage consists of the following steps:

1. *A 10-character password is required to the user. The user password p is hashed using Message-Digest Algorithm 5 (MD5). The hash result H(p) is then directly stored in the token.*
2. *The PTP is extracted as explained in the previous section. As a result of the timing pattern extraction, two dataset are obtained, namely, key pressing pattern and key releasing pattern. A third dataset is created taking the ASCII values of the password characters. These datasets form a universe of 900 possible different combinations. Notice that the PTP may vary even when the biometric information comes from the same user. This is due to the fact that the user is not used to input the chosen password or external factors affect his typing. Then, a training process is needed to overcome these difficulties. The purpose of the training process is to make converge and to generalize the PTP. The training process is as follows:*
   - *The user is required to input ten times the 10-character password chosen. Each time the PTP is extracted as explained previously.*
   - *The 10 PTPs are compared each other point by point. If the two compared points are separated each other for more than 4 time slots when the comparison takes place, that timing pattern is automatically discarded.*
   - *If at least six timing pattern survive this comparison process, the mean is calculated for each point and the result is rounded to the nearest time slot value. Otherwise, the training process must be restarted. Practical experiments showed that a user used to type a password generates the same PTP at least 6 out of 10 tries.*
   - *The resultant PTP obtained from this training process is considered as the global PTP to be used with the proposed approach.*
3. *Three random vectors are generated of 160 values each one. The formed datasets by the key releasing pattern and the ASCII password values are distributed according to the generated RDV which contain pseudorandom values drawn from the standard uniform distribution on the open interval (0, 10). The dataset that correspond to the key pressing pattern is also distributed over generated RDV which contains pseudorandom values drawn from the standard uniform distribution on the open interval (0, 9). Each of the three random vectors corresponds to a coordinate in a 3D plane. Figure 3 shows a 3D random biometric pattern generated using a specific behavioral biometric with a determined random distribution vector. The 3D pattern computed is formed for the resultant eight points obtained of performing the k-medoids algorithm over the random distribution of datasets.*
4. *Once the k-medoids algorithm converges and the 3D pattern is extracted, the pattern, k, is hashed using MD5 and the hash result H(k) is also saved into the token.*
5. *The RDVs used to construct the 3D random biometric pattern are encrypted using the advanced encryption standard (AES) and stored in the token. The MD5 hash*

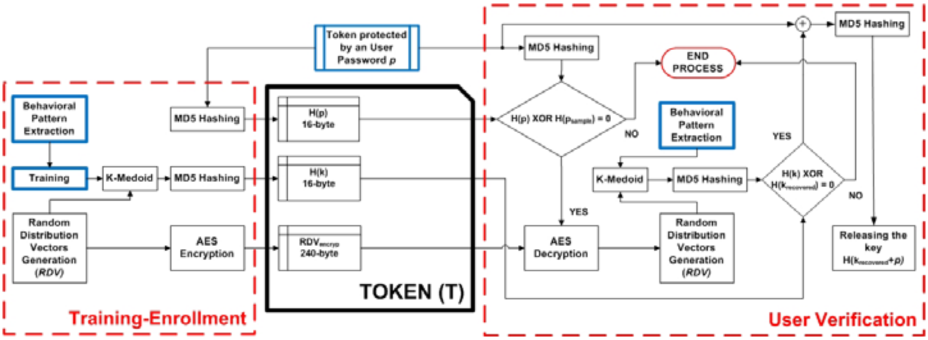

Token protected by an User Password p
Behavioral Pattern Extraction
MD5 Hashing
Training
K-Medoid
MD5 Hashing
Random Distribution Vectors Generation (RDV)
AES Encryption
Training-Enrollment
H(p) 16-byte
H(k) 16-byte
RDV_encryp 240-byte
TOKEN (T)
MD5 Hashing
MD5 Hashing
END PROCESS
H(p) XOR H(p_sample) = 0
NO
YES
Behavioral Pattern Extraction
K-Medoid
MD5 Hashing
H(k) XOR H(k_recovered) = 0
YES
NO
AES Decryption
Random Distribution Vectors Generation (RDV)
Releasing the key H(k_recovered+p)
User Verification

Fig. 2. The three security factor, user password, biometric behavioral sample and token, for the proposed approach

of the user password is used as the 128-bit key that the AES algorithm needs to work. The training-enrollment stage can thus be defined as follows:

$$\langle p, RDV, k\rangle \xrightarrow{\hspace{3em}training - enrollment\hspace{3em}} \begin{bmatrix} H(p) \\ H(k) \\ RDV_{enc} \end{bmatrix} \qquad (3)$$

Now, we proceed to a detailed description of the user verification stage. It must be assumed that the user has the token with the three parameters stored in it.

1. *A user password, p_{sample}, is required to the user who is claiming the identity. Then the password provided for the user is hashed using MD5, $H(p_{sample})$, and compared with the hash stored in the token $H(p)$. If both hashes do not match, the stage ends. Otherwise, the stage continues to step 2.*
2. *To perform AES decryption over the encrypted RDVs stored in the token using as a key the MD5 hash of the password of the user already authenticated.*

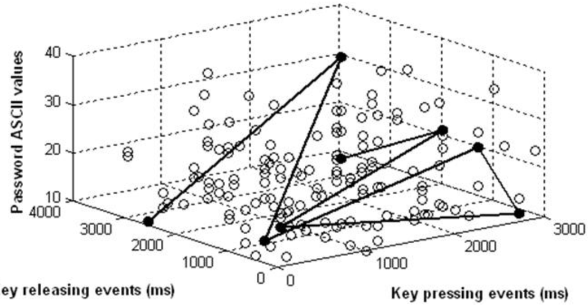


Fig. 3. A random biometric pattern generated using the biometric information with a determined random distribution is shown. The bold line shows the convergence points, 3D pattern, after performing the k-medoids algorithm.

3. *To extract the PTP, of a keystroke dynamics sample presented by the user as described previously.*
4. *To build the 3D random biometric pattern using as datasets the biometric information obtained in step 3 and password in step 1 and as distribution the decrypted RDVs obtained in step 2.*
5. *To apply the k-medoids algorithm over the 3D random biometric pattern built in the previous step to extract the 3D pattern.*
6. *The 3D pattern recovered, $k_{recovered}$, in the previous step is hashed using MD5, $H(k_{recovered})$ and compared to the hash stored in the token H(k). If both hashes do not match, the stage ends. Otherwise, the stage continues to step 7.*
7. *The 3D pattern is added to the ASCII values of the password of the user, $k_{recovered}+p$. The result is hashed using MD5 $H(k_{recovered}+p)$ to obtain a 128-bit random biometric key, k'. This is the cryptographic key that is released and belongs to the verified user. The user verification stage can thus be defines as follows:*

$$\langle p_{sample}, PTP_{sample}, T\rangle \xrightarrow{\hspace{2cm}}_{\textit{user verification}} k' \quad (4)$$

Notice that in both, training-enrollment and user verification, stages is crucial that PTPs, random biometric keys and decrypted RDVs used along the stages must be securely crashed and not retained in memory.

6 Simulation Results

In this section, the performance results of the architecture discussed in the previous sections are reported. To illustrate the performance of the three security factors architecture, a Keystroke Dynamics Database was created. This database contains the timing data of 20 different users. It was collected 20 raw timing data samples total per user without any discretization process. Then, the database contains a global total of 400 timing data to be used to compute the FAR and FRR metrics and the computational cost. The 20 samples collected per user fulfill the training criterion stated previously. Even when the k-medoids algorithm presents several advantages as resistance to noise and outliers compared with other clustering algorithm, it also represents a high computational cost, $Cost(k(n-k)^2)$ due to the fact that it may try every point in the dataset before converging. Table 1 summarizes the maximum, minimum and the mean number of iterations needed to converge using different types of distances. As can be seen, a minimum of 2 iterations are need to make converge the 3D pattern for all distances. However, the Manhattan distance is the most effective distance because it needs at most 4 iterations to converge compared with the 6 iterations that the Euclidean and Chebyshev distance may need or with the 5 iterations that the Minkowski distance evaluated in 3 and 4 may need. Also, the computed mean using the Manhattan distance is the closest value to the minimum number of iterations which assures that the frequency of convergence with 2 iterations is higher compared with the rest of the distances. Then, Manhattan distance is the best choice for the architecture proposed because it needs less iterations to converge and its computational cost, with k=8 and n=160, is considerably lower compared with the rest of the tested distances.

Table 1. Iteration comparison of the k-medoids algorithm working with different types of distance

Distance p	**Maximum**	**Minimum**	**Mean**
1	4	2	2.39
2	6	2	2.44
3	5	2	2.41
4	5	2	2.43
∞	6	2	2.49

Table 2. Performance of FAR and FRR metrics for different time slots

Time slot (ms)	**FAR** (%)	**FRR** (%)
25	2.63	30
50	4.74	15
100	5.26	10
200	10.52	5
300	17.10	5

In training-enrollment stage, it is selected randomly a user then the 20 timing data samples of that user are used in the training process to extract the PTP. Once the PTP has been extracted as explained previously, the proposed architecture generates and stores in the token the information needed in the user verification stage. The FAR and FRR metrics were obtained testing an extracted PTP against the 400 timing patterns stored in the Keystroke Dynamics database. Given that the 20 timing data samples of the user fulfill the training criterion, it may be expected that only the 20 timing data samples that corresponds to the user who is claiming the identity should be accepted as legitimate. The rest, 380 timing data samples of other users, should be rejected by the architecture proposed. However, the FAR obtained in this work is 5.26% because 20 out of 380 timing data samples that do not belong to the user who claims the identity before the proposed architecture were accepted as authentic when they were not. Also, the FRR obtained is 10% because 2 out of 20 timing data samples that in fact belong to the user who claims the identity were rejected even when they represented accurately a timing data sample used to generate the user verification data stored in the token. The FAR is high compared with the combined system reported in [3,4]. However, the FRR has good performance if it is compared to [1,2,4] but it is still high compared to [3]. Table 2 shows how the FAR and FRR metrics are affected by changing the size of the time slot. As the time slot increases the extraction process is less selective this makes that PTPs from different users look similar. This fact then affects FAR negatively. Increasing the size of the time slot affects positively the FRR metric due to the fact that the architecture is able to identify PTP from the same user when the PTP do not differ so much each other.

As can be seen, there is a compromise between the FAR and FRR metric. The size of the time slot must be carefully chosen to save the equilibrium between the metrics. The reason of choosing the 100ms time slot size is due to the fact that the absolute value of the difference of both metric is the minimum among the rest of the data of Table 2 which assures that the uncertainty level is also minimized.

7 Conclusions

In this paper, we have proposed an architecture based on the keystroke dynamics and the k-medoids algorithm. The proposed approach comprises three security factors, namely, user password, behavioral biometric sample and token. It assures that if an attacker compromises at most two factors, he is not going to be able to derive the random cryptographic key. The good performance of the FAR reported in this paper is directly related to the correct selection of the time slot however it is still high compared with the one obtained in systems that combine biometrics and cryptography reported in [3,4]. The FRR has good performance if it is compared with [1,2,4] but it is still not good compared with [3].

The idea behind the three security factor architecture reported in this paper is not limited to work with the PTP as it is extracted here. The extraction technique may be more sophisticated to improve the FAR and FRR and the rest of the approach remain unchanged. Instead of only considering the key pressing pattern and key releasing pattern, it could be added other parameters as the total typing time or the tendency of using certain keys by the user to make even more personal the biometric data. Also, one of the most notable advantages of the proposed approach is that it is not necessary to maintain a centralized database with the biometric information. This fact impacts positively in the social acceptance of the biometric cryptosystems. The proposed three security factor approach is a very secure system because the distribution of the 3D random biometric pattern is randomly generated. Also, if an attacker could compromise the all three factors, the cryptographic key can be easily revoked and renewed by executing the training-enrollment stage again. In the case of that the attacker could somehow derived the cryptographic key, he could compromise the key of that specific user but not the keys of a group or a corporation that could happen in the case of maintaining a centralized database with the biometric information of all users.

References

1. Clancy, T.C., Kiyavash, N., Lin, D.J.: Secure smart card-based fingerprint authentication. In: Proceedings of the 2003 ACM SIGMM Workshop of Biometrics Methods and Application (2003)
2. Goh, A., Ngo, D.C.L.: Computation of cryptographic keys from face biometrics. In: Lioy, A., Mazzocchi, D. (eds.) CMS 2003. LNCS, vol. 2828, pp. 1–13. Springer, Heidelberg (2003)
3. Hao, F., Anderson, R., Daugman, J.: Combining Cryptography with Biometrics Effectively, Computer Laboratory, University of Cambridge, Technical Report Number 640 (2005)

4. Hao, F., Chan, C.W.: Private key generation from on-line handwritten signatures. Information Management & Computer Society 10(4) (2002)
5. Zhang, Q., Couloigner, I.: A New and Efficient K-Medoid Algorithm for Spatial Clustering. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 181–189. Springer, Heidelberg (2005)
6. Barioni, M.C.N., Razente, H., Traina, A.J.M., Traina Jr., C.: Accelerating k-medoid-based algorithms through metric access methods. Journal of Systems and Software 81(3), 343–355 (2008)
7. Kardi, T.: Minkowski Distance of order λ (2009), http://people.revoledu.com/kardi/tutorial/Similarity/MinkowskiDistance.html
8. Garcia-Baleon, H.A., Alarcon-Aquino, V.: Cryptographic Key Generation from Biometric Data Using Wavelets. In: Proceedings of the IEEE Electronics, Robotics and Automotive Mechanics Conference, CERMA 2009 (September 2009)
9. Garcia-Baleon, H.A., Alarcon-Aquino, V., Ramirez-Cruz, J.F.: Bimodal Biometric System for Cryptographic Key Generation Using Wavelet Transforms. In: Proceedings of the IEEE Mexican International Conference on Computer Science, ENC 2009 (September 2009)

A Hardware Architecture for SIFT Candidate Keypoints Detection

Leonardo Chang and José Hernández-Palancar

Advanced Technologies Application Center
7^{th} Ave. ♯ 21812 % 218 y 222, Siboney, Playa, C.P. 12200, Havana City, Cuba
{lchang,jpalancar}@cenatav.co.cu

Abstract. This paper proposes a parallel hardware architecture for the scale-space extrema detection part of the SIFT (Scale Invariant Feature Transform) method. The implementation of this architecture on a FPGA (Field Programmable Gate Array) and its reliability tests are also presented. The obtained features are very similar to Lowe's. The system is able to detect scale-space extrema on a 320 × 240 image in 3 ms, what represents a speed up of 250x compared to a software version of the method.

Keywords: FPGA, SIFT hardware architecture, parallel SIFT.

1 Introduction

In the last few years the use of local features has become very popular due to their promising performance. They have exhibited considerable results in a variety of applications such as object recognition, image retrieval, robot localization, panorama stitching, face recognition, etc.

Almost certainly the most popular and widely used local approach is the SIFT (Scale Invariant Feature Transform) method [5] proposed by Lowe. The features extracted by SIFT are reasonably invariant to image scale, rotation, changes in illumination, image noise, and small changes in viewpoint. This method has been used effectively in all the above mentioned application fields. Lowe divided his method in four major computation stages: i) scale-space extrema detection, ii) keypoint localization, iii) orientation assignment, and iv) keypoint descriptor.

In its first stage, in order to detect scale invariant interest points, Lowe proposed to use scale-space extrema in the Difference-of-Gaussian (DoG) function convolved with the image, which can be computed from the difference of adjacent scale images. To obtain the DoG images several convolutions with Gaussians are produced. This represents a significative computational cost (about 30% of the whole algorithm), which makes it an expensive procedure. Some work has already been done to increase SIFT performance, by using a GPU (Graphic Processor Unit) in PCs [8] or by simplifying the algorithm through approximation [3][4].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 95–102, 2009.

The use of FPGAs (Field Programmable Gate Arrays) is a solution that a large number of researchers has successfully applied to speed up computing applications. Deeper in the SIFT algorithm and specifically in the DoG calculation, it turns out that this task has a great degree of parallelism, making it ideal for implementation in a FPGA. It is mentioned in [7] a system that implements SIFT to aid robotic navigation, which takes 60 ms for a 640 × 480 image. Nevertheless, architecture details or results discussion are not presented. In [6] the most expensive parts of the SIFT algorithm are implemented (i.e. Gaussian Pyramid and Sobel) and some architecture details and algorithm adequacies for hardware are given. This system can run at 60 fps but the image size is not mentioned, neither FPGA area allocation. Another system able to detect SIFT keypoints is presented in [2], which is capable to process 320 × 240 images in 0,8 ms. However, just a few information about the hardware architecture and none from the FPGA area usage are given. A complete implementation is demonstrated in [1], which requires 33 ms per 320 × 240 image.

This paper presents a parallel hardware architecture for one of the most intensive parts of the SIFT algorithm: the scale-space extrema detection. This architecture is implemented in a FPGA, where only 3 ms for scale-space extrema detection on a 320 × 240 sized image are required.

The organization of this paper is as follows: In Section 2 the target algorithm is explained. In Section 3 some issues of the scale-space extrema detection are discussed, which are later used by the architecture proposed in Section 4. In Section 5, implementation details and reliability tests for our system are presented. The work is concluded in Section 6.

2 Detection of Scale-Space Extrema in the SIFT Method

For a given image $I(x, y)$, the SIFT detector is constructed from its Gaussian scale-space, $L(x, y, \sigma)$, that is built from the convolution of $I(x, y)$ with a variable-scale Gaussian: $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$, where $G(x, y, \sigma)$ is a Gaussian kernel and $*$ is the convolution operator in x and y. The Gaussian scale space is created by generating a series of smoothed images at discrete values of σ. Thus the σ domain is quantised in logarithmic steps arranged in O octaves, where each octave is further subdivided in S sub-levels. The value of σ at a given octave o and sub-level s is given by: $\sigma(o, s) = \sigma_0 2^{o+s/S}, \quad o \in [0, ..., O-1], \quad s \in [0, ..., S-1]$, where σ_0 is the base scale level, e.g., $\sigma_0 = 1.6$. At each successive octave the data is spatially down-sampled by a factor of two.

To efficiently detect stable keypoint locations in scale space, Lowe proposed using scale-space extrema in the DoG scale-space, $D(x, y, \sigma)$, computed from the difference of adjacent scales: $D(x, y, \sigma(o, s)) = L(x, y, \sigma(o, s+1)) - L(x, y, \sigma(o, s))$.

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each pixel in the DoG images is compared to its eight neighbors at the same image, plus the nine corresponding neighbors at adjacent scales. If the pixel is larger or smaller than all these neighbors, it is selected as a candidate keypoint.

3 The Proposed Parallel Detection of Scale-Space Extrema

The main motivation for the use of FPGAs over conventional processors is given by the need to achieve higher performance, better tradeoff cost-benefits and scalability of a system. This is possible thanks to the inherent parallelism in these devices, which by their physical characteristics, is able to keep all operations activated. Therefore, to achieve such profits and a significant speedup in the detection of scale-space extrema, is essential to exploit the parallelism of this algorithm. Nevertheless, there are other factors to consider in a FPGA design such as area and power requirements. Hence, this algorithm must be rewritten to take advantage of the parallel structure afforded by implementation in hardware, taking into account area and power requirements.

3.1 Exploiting Data Parallelism

Convolution is one of the most expensive operations that are used in image processing applications and particularly in the SIFT method. Then, it is an important issue to deal with.

If I is a two-dimensional image and g is a convolution mask of odd size $k \times k$, then the convolution of I and g is defined as:

$$f(x,y) = \sum_{-i}^{i} \sum_{-j}^{j} I(i,j) g(x-i, y-j), \text{where } i,j = \lfloor \frac{k}{2} \rfloor. \qquad (1)$$

As can be seen in (1), for the calculation of $f(x_1, y_1)$ only a neighborhood in I of size $k \times k$ with center (x_1, y_1) is needed. Therefore, in 2D convolution a high potential for data parallelism is available, specifically of SPMD (Single Process, Multiple Data) type.

3.2 Exploiting Separability Property of the Gaussian Kernel

With the aim of reducing the number of arithmetic operations, the separability and the symmetry properties of the Gaussian are considered.

A 2D filter kernel is separable if it can be broken into two 1D signals: a vertical and a horizontal projection. The Gaussian kernel could be separated as follows:

$$G(x,y,\sigma) = h(x,\sigma) * v(y,\sigma),$$

where

$$h(x,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}, \text{and } v(y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-y^2/2\sigma^2}$$

Thus, the 2D convolution can be performed by first convolving with $h(x,\sigma)$ in the horizontal direction, and then convolving with $v(y,\sigma)$ in the vertical direction. 1D convolution, to compute a value of the output, requires k MAC operations. However, as is described in (1), 2D convolution in spatial domain requires k^2 MAC (multiplication and accumulation) operations. Therefore, the computational advantage of separable convolution versus nonseparable convolution is $k^2/2k$.

3.3 Octaves Processing Interleaving

As stated in Section 2, at each successive octave, the image size is downsampled by a factor of two by taking every second pixel in each row and column, i.e. $I_o(x, y) = I_{o-1}(2x, 2y)$. After downsampling by a factor of two, the total number of pixels is reduced by four. In hardware, to reduce the size of the data, its sample rate is reduced by the same factor. If at each successive octave the data size is reduced by four, the sample period τ of an octave o is given by

$$\tau(o) = \tau_0 4^o, \tag{2}$$

where τ_0 is the first octave sample period. Consequently, after subsampling, there is a large percentage of idle processing time $\hat{\imath}$ in respect of the first octave sample period, which is defined by $\hat{\imath} = \frac{\tau(o)-1}{\tau(o)}$.

This idle processing gap makes feasible the processing of the O octaves of a scale in a single convolution processor. This could be possible by interleaving the O convolution processes so that for all the octaves at a given time t the number of processed elements $p(o, t)$ satisfies that

$$p(o, t) = \left\lfloor \frac{t + \varepsilon(o)}{\tau(o)} \right\rfloor, \tag{3}$$

where $\varepsilon(o)$ is the delay of octave o in the interleaving line.

Here, for the O octaves interleaving is assumed that the first octave sample period is equal or greater than two clock cycles, if not, $O - 1$ octaves are interleaved and τ_0 would be the second octave sample period.

4 The Proposed Hardware Architecture

In the architectures proposed in [6] and [1], it is used one convolution block per each convolution operation, dividing the processing by octaves and resulting in $O \cdot S$ convolution blocks. In this work we present an architecture that only uses S convolution blocks for the $O \cdot S$ convolution operations, dividing the processing by scales and providing the same throughput.

A block diagram of the overall architecture is shown in Figure 1 a). This diagram shows a system of four octaves, five scales and a seven coefficient kernel ($O = 4, S = 5, k = 7$); but it could be generalized for any configuration.

The hardware architecture consists, in a major manner, of scale computation blocks (SCB). One SCB performs the O Gaussian filtering operations of a given scale as discussed in Section 3.3. Therefore, each SCB has O input and output ports, one for each octave respectively, where the sample period of each octave is defined by equation (2).

As can be seen in Figure 1 a), the SCB blocks are interconnected in cascade in order to have a constant convolution kernel size and to avoid convolutions with big kernel sizes.

A SCB block, to perform Gaussian filtering, takes advantage of the separability property of the Gaussian kernel as described in Section 3.2. As can be seen

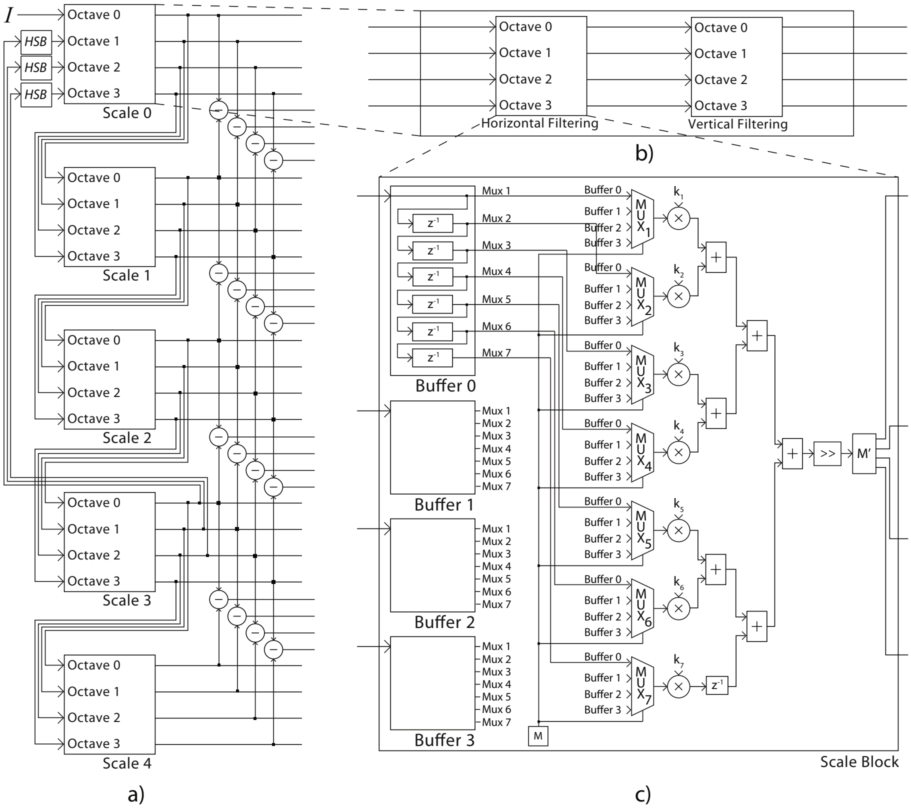


Fig. 1. Proposed pipelined architecture

in Figure 1 b), this block performs Gaussian filtering by convolving an image in the horizontal direction and then in the vertical one.

The internal arrangement of the horizontal filter in the SCB block is detailed in Figure 1 c). Each input signal is shifted throughout k registers, where k is the convolution kernel width. The k values of the O octaves are multiplexed with the aim of controlling the octaves processing order and accomplishing octaves processing interleaving. The multiplexers logic for octaves interleaving at a given time t is determined by the M block which implements function $m(t)$ and fulfills the condition stated in (3). The interleaving order is defined as follows:

$$m(t) = \begin{cases} o_0 & \text{if } t \equiv \varepsilon(o_0) \mod \tau(o_0) \\ o_1 & \text{if } t \equiv \varepsilon(o_1) \mod \tau(o_1) \\ \vdots & \vdots \\ o_{O-1} & \text{if } t \equiv \varepsilon(o_{O-1}) \mod \tau(o_{O-1}). \end{cases}$$

The structure of the vertical filter is the same as the horizontal; with the distinction that each buffer stores the last k lines instead of the last k elements.

By interleaving octaves processing in a single convolution processor it is possible to save a lot of silicon area and the consequent power consumption reduction. Also, to avoid operating with fixed point values, kernel coefficients are multiplied by an appropriate constant. Later, the filtered result is normalized by dividing it by this same constant. More desirably, the constant chosen must be a power of two in order to replace the division operation by a simple shift.

The HSB block in Figure 1 a) performs image downsampling.

5 FPGA Implementation and Experimental Results

5.1 Implementation Characteristics

A system configured with $O = 4$, $S = 6$ and $k = 7$ to process 320×240 sized images was implemented on a Xilinx Virtex II Pro FPGA (XC2VP30-5FF1152). This system was implemented using System Generator + Simulink. The estimated resources occupied by this implementation and its comparison with Bonato et al. system [1] are summarized on Table 1. As discussed in previous sections, the system returns a result every two clock cycles. Under this implementation, with a 50 MHz clock rate, the time taken to detect scale-space extrema in a 320×240 image is 3 ms, so, it is possible to process 330 frames per second. This result was compared, in terms of performance, with Vedaldi software implementation [9] running on a PC (1.8 GHz Core Duo and 1 GB RAM). Our system proved a significative speed up of 250x.

Table 1. Implementation Characteristics and Comparison with [1]

Resources	Our System $O = 4$, $S = 6$ and $k = 7$	DoG part of [1] $O = 3$, $S = 6$ and $k = 7$
Slices	5068	-
Flip-flops	6028	7256
Look Up Tables	6880	15137
Blocks RAM	120 (2.1 Mb)	0.91Mb

5.2 System Reliability

In order to test our implementation reliability, we checked for matches between features found by a complete software version and a hybrid implementation where scale-space extrema were obtained by our system. The SIFT software implementation used was Lowe's [5]. The hybrid implementation was created from Vedaldi's, where the first SIFT computation stage was executed by our system. We looked for matches between these two implementations on 26 images. The main differences between matches are due to the approximations on the DoG calculation process. However, these approximations did not greatly affected the final detected features. The mean errors in coordinates, scale and orientation of the detected features are $\Delta x = 1.127$, $\Delta y = 1.441$, $\Delta\sigma = 0.149$ and $\Delta\theta = 0.047$ respectively.

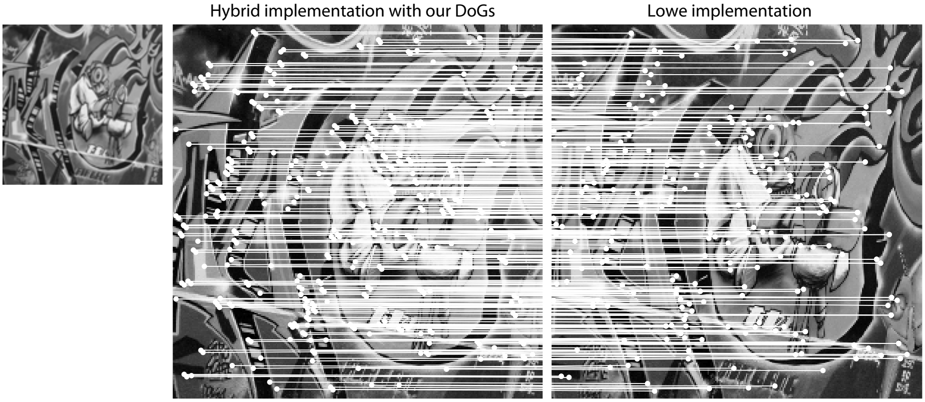


Fig. 2. Matches between the hybrid implementation (using our system results) and Lowe implementation for an example image

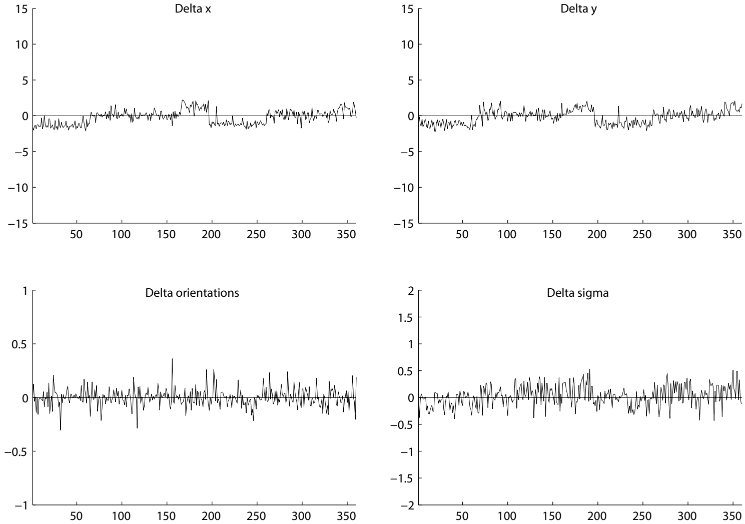


Fig. 3. Errors in coordinates, scale and orientation of the detected features for an example image

Figure 2 shows an example of detected features by the two implementations and their matches. The errors in coordinates, scale and orientation computation for this example image, are shown in Figure 3.

6 Conclusions

We have proposed a parallel hardware architecture for one of the most computationally expensive parts of the SIFT algorithm: the scale-space extrema detection. For this purpose, we exploited some algorithm particularities such as its intrinsic data parallelism, the separability property of the Gaussian kernel and the octaves processing interleaving possibility. The mean errors of the SIFT features detector and descriptor, based on our system results, are $\Delta x = 1.127$, $\Delta y = 1.441$, $\Delta\sigma = 0.149$, $\Delta\theta = 0.047$. The results of the comparisons showed that our system needs less silicon area than Bonato et al. system, even processing one more octave. This area profit is due more to octaves processing interleaving.

References

1. Bonato, V., Marques, E., Constantinides, G.A.: A parallel hardware architecture for scale and rotation invariant feature detection. IEEE Transactions on Circuits and Systems for Video Technology 18(12), 1703–1712 (2008)
2. Djakou Chati, H., Muhlbauer, F., Braun, T., Bobda, C., Berns, K.: Hardware/software co-design of a key point detector on FPGA. In: FCCM 2007: Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Washington, DC, USA, pp. 355–356. IEEE Computer Society, Los Alamitos (2007)
3. Grabner, M., Grabner, H., Bischof, H.: Fast approximated SIFT. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 918–927. Springer, Heidelberg (2006)
4. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513 (2004)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
6. Pettersson, N., Petersson, L.: Online stereo calibration using FPGAs. In: Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, pp. 55–60 (2005)
7. Se, S., Ng, H.k., Jasiobedzki, P., Moyung, T.j.: Vision based modeling and localization for planetary exploration rovers. In: 55th International Astronautical Congress 2004 (2004)
8. Sinha, S., Frahm, J.-M., Pollefeys, M., Genc, Y.: Feature tracking and matching in video using programmable graphics hardware. Machine Vision and Applications
9. Vedaldi, A.: An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD (2007)

Analysis of Non Local Image Denoising Methods

Álvaro Pardo

Department of Electrical Engineering, Faculty of Engineering and Technologies, Universidad Católica del Uruguay
apardo@ucu.edu.uy

Abstract. Image denoising is probably one of the most studied problems in the image processing community. Recently a new paradigm on non local denoising was introduced. The Non Local Means method proposed by Buades, Morel and Coll attracted the attention of other researches who proposed improvements and modifications to their proposal. In this work we analyze those methods trying to understand their properties while connecting them to segmentation based on spectral graph properties. We also propose some improvements to automatically estimate the parameters used on these methods.

1 Introduction

Image deonising is probably one of the most studied problems in image processing. The main goal of denoising is to remove undesired components from the image. These undesired components, usually defined as noise, can be of different nature: random noise introduced at acquisition time, noise introduced during transmission, noise due to degradation such in films, etc. In this work we assume that the observed image, x, is the result of adding a random noise component n to the original noiseless image z. Therefore, the relationship between those images at pixel i becomes: $x_i = z_i + n_i$.

The problem of image denoising then is to estimate z while preserving its features such as edges and texture. There is usually a tradeoff between noise reduction and feature preservation. Since image features usually involve high frequencies linear low pass filters usually produce poor results regarding feature preservation. For this reason several non linear or locally adapted methods have been developed. As examples we mention median filters, anisotropic diffusion and wavelet thresholding. More recently non local methods attracted the attention of the image processing community. Starting from the pioneering work of Efros and Leung [7] several non local methods have been introduced for image denoising. In [5] Buades, Morel and Coll presented the Non Local Means (NLM) denoising method. The underlying idea of this method is to estimate, z_i, using a weighted average of all pixels in the image. Given the pixel to be denoised, i, the weights w_{ij} measure the similarity between neighborhoods centered at i and j. The trick is that corresponding neighborhoods are found all along the image imposing a non local nature to the method. Similar methods can be found in [1,3,8]. A review of several denoising strategies and its comparison against non local means can be found in [6] and [4].

In this work we study the behavior of non local denoising methods. First we show the connection of non local means to graph clustering algorithms and use it to study the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 103–110, 2009.

denoising performance. Using synthetic images we will show the limitations of standard non local means and propose an improvement to automatically estimate the parameters of NLM based on noise variance estimation.

2 Non Local Means Denoising

The NLM algorithm [5] estimates the denoised value at pixel i using a weighted average of all pixels in the image:

$$\hat{x}_i = \sum_j \bar{w}_{ij} x_j$$

The weights $\bar{w}_{ij}$ reflect the similarity between pixels i and j based on the distance between neighborhoods around them (see equations (1) and (2)).

Ideally, due to the non local nature of the algorithm, similar neighbors are found across the whole image. This has two drawbacks. The first one is the computational complexity of searching similar neighborhoods across the whole image. The second one is related with the fact that taking weighted averages for all pixels in the image does not achieve the best MSE score for this algorithm. This issue was addressed in [2] and [6] noted the problems with edge pixels. The problem is that in some cases the weights w_{ij} are not able to discriminate between different neighborhoods classes. This is especially the case along edges since pixels along them have less corresponding neighborhoods in the image. Other authors that addressed the computational complexity of NLM encountered this trade off, for instance see [8]. Based on these considerations we can see that a better solution is obtained via averaging only pixels within the same class of neighborhoods. Therefore, the denoising performance depends in a good neighborhood classification. In what follows we will review NLM and show its connection with segmentation based on spectral clustering.

To conclude this discussion we point out that the performance of NLM depends on the selection of the parameter σ. Although in [6] the authors provide some guidance on how to select its value, we will show that the selection of σ has a great impact on the results.

2.1 Graph Formulation of NLM

Let x_i be the original noisy image value at pixel i. Its denoised version using NLM can be obtained as [5]:

$$\hat{x}_i = \frac{\sum_j w_{ij} x_j}{\sum_j w_{ij}} \tag{1}$$

where the weights w_{ij}[1] are computed using a gaussian kernel,

$$w_{ij} = \exp(-||N_i - N_j||^2/\sigma^2) \tag{2}$$

and N_i, N_j are image patches of size $(2K+1) \times (2K+1)$ centered at pixels i and j.

[1] $\bar{w}_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}$.

The equation (1) can be rewritten in matrix notation as follows. Let the matrix W be the one with entries w_{ij}, and D the diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$. If we consider $\mathbf{x}$ as the vectorial version of the image, scanned in lexicographic order, equation (1) can be rewritten as:

$$\hat{\mathbf{x}} = D^{-1}W\mathbf{x} \tag{3}$$

The matrix $L = D^{-1}W$ defines an operator which filters the image, $\mathbf{x}$, to obtain a denoised version $\hat{\mathbf{x}}$. This denoising filter is an image adapted lowpass filter since the operator depends on the image itself. Therefore, the properties of the matrix L determine the denoising result. If we are interested in the properties of a denoising algorithm it is of common use to study the properties of the residual after denoising, $\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}}$. If we write the residual using equation (3) we obtain: $\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - D^{-1}W\mathbf{x} = (Id - D^{-1}W)\mathbf{x}$. The matrix $H = (Id - D^{-1}W)$ is the highpass operator associated with the lowpass operator defined by matrix L.

If we view pixels x_i as nodes of a graph connected with weights w_{ij} the matrix H is the normalized Laplacian of the graph which is used in Normalized Cuts (NC). In [10] Malik and Shi presented a relaxed version of the normalized cut which solution is the second eigenvector of H. In this way we show the connection between NLM and segmentation based on NC.

Matrices L and H share the same eigenvectors; if φ_k is an eigenvector of L with eigenvalue λ_k then φ_k is an eigenvector of H with eigenvalue $1 - \lambda_k$. From these considerations we conclude that the eigenvectors and eigenvalues of L and H play an important role in the denoising process.

It can be shown that the multiplicity of the eigenvalue with value one of L corresponds to the number of connected components in the graph [11]. These connected components correspond in our case to the neighborhood classes. So, since an ideal denoising method should average only points in the same classes, the spectrum of the graph related to L is important to measure the performance of the algorithm. We will use the multiplicity of the eigenvalue one to judge the performance of our proposal and compare it with traditional NLM.

3 Experiments with Synthetic Images

To study the denoising performance of NLM we will use a synthetic image with three regions with values 1, 3 and 1 plus Gaussian noise with variance 0.3 (see Figure 2). We consider patches of size 3 × 3 which gives us six different noiseless neighborhood configurations as show in Figure 2.

Following the same idea proposed in [2] we applied NLM together with a restriction on the number of neighboring patches used for the denoising process. That is, for each pixel to be denoised we considered only pixels with neighborhood similarity greater than ε, that is $\exp(-||N_i - N_j||^2/\sigma^2) \geq \varepsilon$, and computed the error for different values of ε. For this experiment we set $\sigma = 12\sigma_n$ as suggested in [6]. We also computed the error over each region of the image. That is, based on the local configurations show in Figure 2, we segmented the image in six regions and computed the denoising error for each one of them. The results of these simulations are show in Figure 1. As we can see

Table 1. Minimum errors for regions from image in Figure 2

Region	ε	Error NLM	Type
1	0.60	0.0011	non-boundary
2	0.80	0.0036	boundary
3	0.80	0.0099	boundary
4	0.60	0.0005	non-boundary
5	0.75	0.0187	boundary
6	0.75	0.0069	boundary

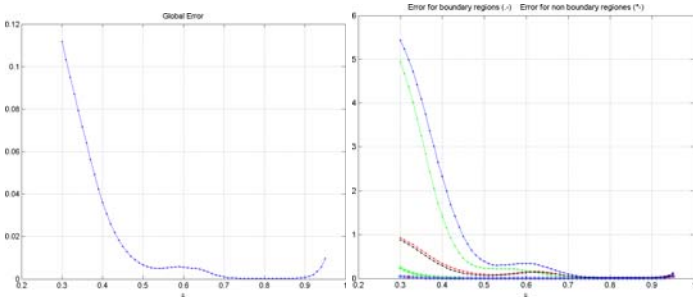


Fig. 1. Left: Global error evolution. Right: Region error evolution.

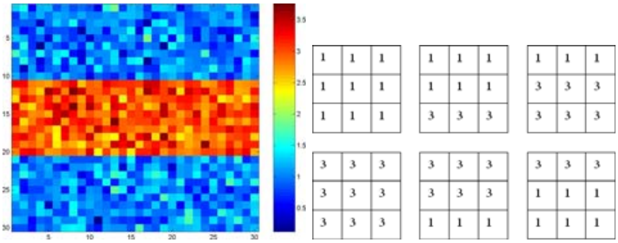


Fig. 2. Left: Noisy image. Right: Neighborhood configurations.

the global error has a U shaped curve. The error decreases as ε increases which means that the error improves while we restrict the set of neighborhoods used. Also as ε goes to one the error increases as the number of points used for the estimation decreases. In the middle we obtain the minimum global error which is quite stable. This means that considering all neighborhoods for the denoising process is clearly not the best option. To understand the reasons of this behavior we computed the errors per region shown in Figure 1. It is clear that boundary regions (neighborhoods with pixels of two regions) perform differently than non boundary regions (neighborhoods with pixels of the same region). Non boundary regions have an almost constant error while boundary regions show a stronger dependence on ε. This explains the obtained global error. In Table 1 we show the minimum errors per regions and the values of ε where these minima are achieved. In next section we will use these results to design an improved NLM.

4 Modified NLM

In this section we address the automatic estimation of the parameters σ of NLM and ε as discussed earlier and present a modified NLM (MNLM).

4.1 Parameter Estimation

Noise variance estimation The estimation of σ will be based on the noise variance. For Gaussian noise the estimation of its variance can be done applying methods as the ones proposed in [9].

Estimation of $\boldsymbol{\sigma}$. Following [6] we set σ proportional to the noise variance: $\sigma = h\sigma_n$. We propose to choose the value of h looking at the expected distances for neighborhoods inside the same class. The expected squared distance for two identical neighborhoods corrupted by Gaussian noise with zero mean and variance σ_n is:

$$\bar{d}^2 = E\{||N_i - N_j||^2\} = E\left\{\sum_{k=1}^{(2K+1)^2} (x_i^k - x_j^k)^2\right\} \tag{4}$$

$$= \sum_{k=1}^{(2K+1)^2} E\left\{(n_i^k - n_j^k)^2\right\} = 2(2K+1)^2\sigma_n^2. \tag{5}$$

We set the value of h in order to obtain weights greater than γ for similar neighborhoods. In this way the value of h is defined as the one that satisfies the following equation:

$$\exp\left(\frac{-\bar{d}^2}{h^2\sigma_n^2}\right) = \gamma.$$

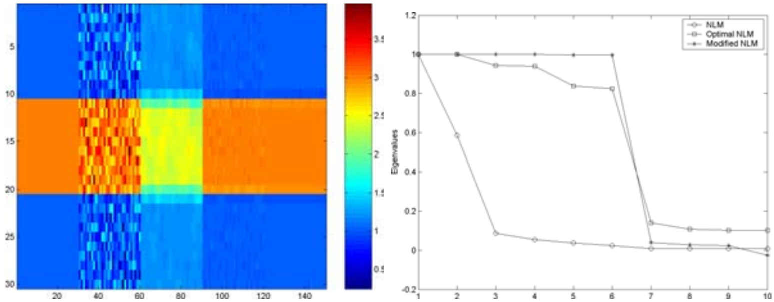
If we substitute $\bar{d}^2$ in previous equation we obtain:

$$h = \sqrt{\frac{2(2K+1)^2}{\log(1/\gamma)}}$$

Finally we have to select the value for ε. As we said before better results are obtained when only neighborhoods with similarities greater than ε are considered. Therefore, we let $\gamma = \varepsilon$. So, instead of parameter σ we have a new parameter ε which controls the neighborhoods considered in the estimation and the value of σ. Also this parameter does not depend on the input image but only on the noise level estimation. In order to consider only neighborhoods of the same class ε must take values close to one. In following sections we will analyze this proposal at the light of the relationship between NLM and segmentation and show that taking $\varepsilon = 0.8$ gives excellent results for a set of real images.

Table 2. Global MSE scores

	NLM	Best NLM	Modified NLM
MSE	0.2046	0.0012	0.0006



(a) (b)

Fig. 3. (a) Fron left to right: noiseless image, noisy image, result of NLM, result of best NLM and result of modified NLM. (b) Eigenvalues.

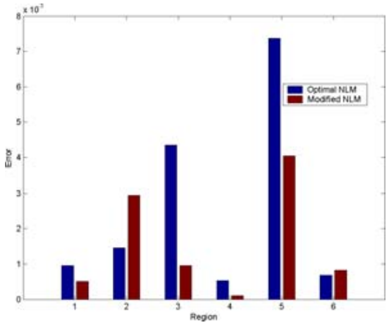


Fig. 4. NLM against MNLM: Errors per region

4.2 Modified NLM and Graph Cuts

In this section we will compare the performance of MNLM against the original NLM using the results from section 3 and the image showed in Figure 2. The image in Figure 2 was filtered with three algorithms: the original NLM with $\sigma = 12\sigma_n$, MNLM with $\varepsilon = 0.8$ and the best NLM in which case we selected the parameter $\sigma = 5\sigma_n$ that gives the smallest global MSE. The obtained MSE errors are presented in Table 2. In Figure 3(a) we show the original noiseless image, the noisy image and the images corresponding to the methods in evaluation. As we can see the original NLM gives the worst result in terms of MSE and visual quality while the modified NLM obtains the best overall performance (see MSE scores in Table 2). If we look at MSE per region we can see in Figure 4 that MNLM performs better than NLM in four out of six of the regions.

Finally we present the eigenvalues of the corresponding matrices L for each method. In Figure 3(b) we show the first ten eigenvalues for each method. It is clear that MNLM has better performance since it has six eigenvalues of value one corresponding to the six regions present in the image. We recall that the multiplicity of the eigenvalue one corresponds to the number of connected components in the graph, i.e. the number of neighborhood classes which in this case is six.

4.3 Results for Real Images

Here we compare the best performances of NLM and our modified NLM (MNLM). Each of the images in Table 3 was contaminated with independent and additive Gaussian noise with $\sigma_n = 10$. We used neighborhoods were of 3×3 and to reduce the computational complexity we used a search window of 21×21. For the evaluation of the results we use Mean Square Error (MSE) and the Structural Similarity Index (SSIM) proposed in [12] which compares the similarity between images using perceptual factors.

With this results we confirm that NLM attains the best result at $h = 3$ in all cases. This contrast with the values of h selected by Buades, Morel and Coll in [5,6] where they suggest $h \in [10, 15]$. Clearly with their selection for h the results are not the best possible. We confirm this based on MSE and SSIM. We must stress that in all cases the optimum is achieved with the same value of h. On the other hand, the results obtained with our modified NLM method present similar results as the ones given by NLM. Therefore based only on MSE and SSIM we cannot say which method is better. As for MNLM the best score values are obtained with $\epsilon = 0.8$ in all case but one.

Result Analysis. To conclude the evaluation we give an explanation on why the best results of NLM are similar to the ones of MNLM. In previous experiments the parameter h which gives the best results for NLM is in all cases 3. The difference between both methods is the width of the Gaussian kernel. For modified NLM the width is $\sigma^2_{MNLM} = \frac{2(2K+1)^2}{\log(1/\gamma)}$ and for NLM the width which produces the best results is $\sigma^2_{NLM} = 3^2\sigma^2_n$. The other difference is that for MNLM we consider only weights above $\varepsilon = \gamma$ and for NLM we consider all weights. The distances for which MNLM gives weights γ are $d^2_\gamma = 2(2K+1)^2\sigma^2_n$. If we substitute this distance in the NLM kernel we get: $\exp(-2(2K+1)^2/3^2) \approx 0.13$. Therefore the corresponding weights for NLM are small and explain the similarity between results of NLM and MNLM. We confirmed the same results using neighborhoods of size 5×5 but due to the lack of space we can not report them here.

Table 3. Minimum errors for regions from image in Figure 2

	NLM			MNLM		
Image	h*	MSE*	SSIM*	ε*	MSE*	SSIM*
Barbara	3	30.11	0.913	0.75	28.93	0.922
Baboon	3	63.45	0.895	0.80	70.67	0.890
Couple	3	35.13	0.883	0.80	35.97	0.884
Einstein	3	35.03	0.859	0.80	36.01	0.858
Goldhill	3	33.78	0.868	0.80	34.61	0.869

5 Conclusions

In this work we study the relationship between non local denoising methods and spectral graph properties. Based on these results we proposed a modification of NLM which automatically estimates the parameter σ. We justified the proposed algorithm using the connection between NLM and graph clustering. Based on simulations we showed that this approach outperforms the NLM with the parameters suggested by Buades and colleagues in [5]. Furthermore we showed that this parameter setting is the best one for all images tested when comparing the MSE scores.

References

1. Awate, S.P., Whitaker, R.T.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. IEEE Trans. Pattern Anal. Mach. Intell. 28(3), 364–376 (2006)
2. Bertalmio, M., Caselles, V., Pardo, A.: Movie denoising by average of warped lines. IEEE Transactions on Image Processing 16(9), 2333–2347 (2007)
3. Boulanger, J., Kervrann, Ch., Bouthemy, P.: Adaptive space-time patch-based method for image sequence restoration. In: Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP 2006) (May 2006)
4. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighborhood filters and its solution. IEEE Transactions on Image Processing 15(6), 1499–1505 (2006)
5. Buades, A., Coll, B., Morel, J.M.: Denoising image sequences does not require motion estimation. In: Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, pp. 70–74 (2005)
6. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. SIAM Multiscale Modeling and Simulation 4(2), 490–530 (2005)
7. Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: IEEE Int. Conf. on Computer Vision, ICCV 1999, pp. 1033–1038 (1999)
8. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhodds. IEEE Signal Processing Letters 12(12), 839–842 (2005)
9. Olsen, S.I.: Noise variance estimation in images. In: Proc. 8th SCIA, pp. 25–28 (1993)
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
11. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4) (2007)
12. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Processing 13(4), 600–612 (2004)

III Segmentation, Analysis of Shape and Texture

Texture Characterization Using a Curvelet Based Descriptor

Francisco Gómez and Eduardo Romero

Bioingenium Research Group - Faculty of Medicine
University of Colombia, Bogotá DC - Colombia

Abstract. Feature extraction from images is a key issue in image classification, image representation and content based image retrieval. This paper introduces a new image descriptor, based on the curvelet transform. The proposed descriptor captures edge information from the statistical pattern of the curvelet coefficients in natural images. The image is mapped to the curvelet space and each subband is used for establishing the parameters of a statistical model which captures the subband marginal distributions as well as the dependencies across scales and orientations of the curvelet. Finally, the Kullback−Leibler distance between the statistical parameters is used to measure the distance between images. We demonstrate the effectiveness of the proposed descriptor by classifying a set of texture images, and with a simple nearest neighbour classifier we obtained an accuracy rate of 87%.

Keywords: texture characterization, curvelet transform, generalized Gaussian distribution, Kullback−Leibler distance.

1 Introduction

The capacity of a mapping to generate features with discriminant characteristics in textures is of paramount importance for the problem of classification and/or retrieval. Typical applications include microscopical or satellite images [1]. Formally, the feature extraction process is thought of as a mapping of an image collection to a characteristic space, which provides a representation where similar images are close and different images are far; this property is known as the discriminating space power. Images projected onto this space are characterized by features which capture some properties of the image, typically some statistical properties from the data. Likewise, a metric for the space is also needed. In the particular case of textures, the most popular characteristic spaces are currently the wavelets, Gabor and DCT transforms [2]. Unfortunately, these spaces are sub-optimal for this problem because textures are naturally entailed with geometrical, scale and directional properties which are poorly described with these transforms [3]. Some of the features already used for this problem capture information of the energy coefficient distribution and include the total energy, the mean and the variance [2]. However, these features do not reflect correctly the statistical properties of natural images [4]. Finally, the usual metrics includes Euclidian or distances between probability density functions such as the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 113–120, 2009.

Kullback−Leibler [5]. In these terms the problem of texture characterization consists in constructing a feature with high discriminative power that takes into account the statistical image contents.

The problem of texture characterization with curvelets was already addressed by Semler [6], who studied the performance of several characteristics, namely: the energy, entropy, mean and standard deviation of the curvelet subbands. Results showed significant improvement when comparing with wavelets, but this characterization did not take into account the particular statistical patterns of the curvelet coefficients in texture images [7]. Sumana [8] also proposed the curvelet subband mean and variance as features while the Euclidian distance between subbands measured closeness. Results showed again improvement when comparing with Gabor features. However, texture curvelet subbands are not described by simple Gaussians so that mean and variance result insufficient to describe the observed distribution [7].

In this paper we present a new global descriptor, entailed with the previously described properties. The curvelet space is used to capture information about edges which is in fact one of the most discriminating features [9]. These features are the moments of a generalized Gaussian density (GGD) which provides a good approximation to the marginal curvelet subband distribution [7], whilst the Kullback−Leibler distance measures differences between curvelet coefficient distributions. A main contribution of this paper is to demonstrate that taking into account an entire statistical characterization of the curvelet coefficient, results in a highly discriminative, precise and simple descriptor of natural textures. The rest of this paper is organized as follows: Section materials and methods introduces the new feature, Section Results demonstrates the effectiveness of this descriptor in classification tasks. Finally, the last section concludes with a discussion and future work.

2 Materials and Methods

The inputs are two images which are curvelet-represented. Frequency subbands are statistically characterized using the moments of a GGD and finally a Kullback-Leibler divergence computes the distance between the two representations. This strategy will be further explained hereafter:

2.1 The Curvelet Transform

The curvelet transform is a multiscale decomposition [10], developed to naturally represent objects in two dimensions, improving the wavelet limitations in 2D. Curvelets are redundant bases which optimally represent 2D curves. Besides the usual information about scale and location, already available from a wavelet, each of these frame elements is able to capture information about orientation while also fulfills the parabolic anisotropic scaling law $width \approx length^2$, whereby curves at different scale levels conserve their geometrical relationships [10]. A curvelet can be thought of as a radial and angular window in the

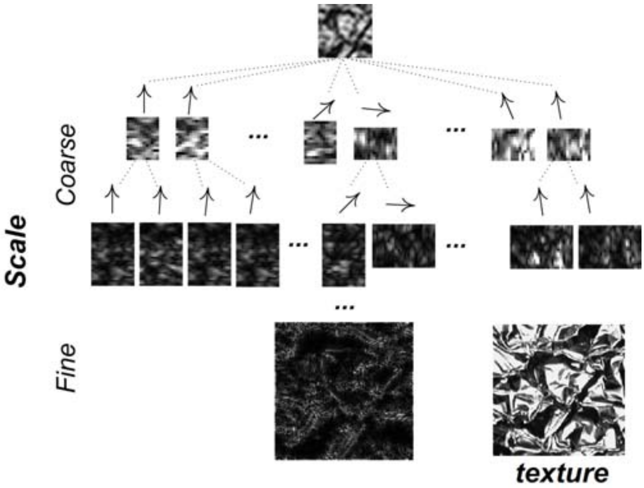


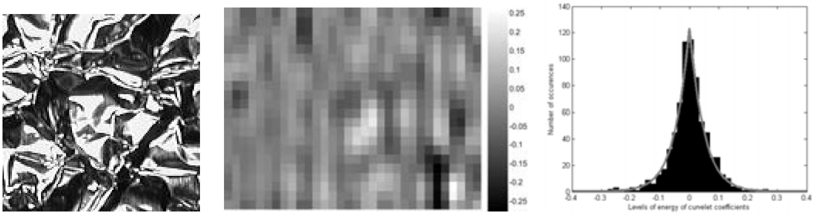
Fig. 1. The figure illustrates a curvelet decomposition of a texture: from top to bottom, increasing levels of detail, from left to right, different orientations

frequency domain, defined in a polar coordinate system. This representation is constructed as the product of two windows: the angular and the radial dyadic frequential coronas. The angular window corresponds to a directional analysis, i.e., a Radon transform, and the radial dyadic window is a bandpass filter whose cut frequencies extract the image information that follows the parabolic anisotropic scaling law [10]. Curvelet bases were designed to fully cover the frequency domain, in contrast to other directional multiscale representations such a the Gabor transform [11], with which some information is always lost. Thanks to the anisotropic scale, curvelets adapt much better to scaled curves than Gabor transform, improving the representation at different scales and noise robustness [11]. All these statements have been experimentally demonstrated by comparing wavelets, curvelets and Gabor in classification and retrieval tasks [8].

The curvelet $\varphi_{j,l,k}$ is indexed by scale j, orientation l and position k, and the curvelet coefficient is simply $c_{j,l,k} = \langle f, \varphi_{j,l,k} \rangle$, that is to say the projection of the image f over the curvelet basis $\varphi_{j,l,k}$. Typically, the spatial curvelet coefficients with the same scale and orientation are grouped per subbands. The figure 1 shows a curvelet multiscale decomposition example.

2.2 Statistical Characterization

Psychophysical research has demonstrated that two homogeneous textures are not discriminable if their marginal subband distributions are alike [9]. This fact suggests that these distributions have a highly descriptive capacity, at least for the texture problem. This discriminative power was also experimentally verified for Wavelet and Gabor representations [2]. In the curvelet case, each subband contains information about the degree of occurrence of similar curves within the



(a) Texture. (b) Curvelet Subband (c) Curvelet histogram.

Fig. 2. Curvelet histogram example (scale 3 and orientation 16)

image, i.e., edge energy levels with similar direction and size. Figure 2 shows a typical example of the curvelet coefficient histogram of an image subband. The kurtosis in this case is about 7.4 so that a Gaussian density is not enough as to match the observed energies. Therefore, the mean and variance calculated from a Gaussian, used in a previous works [6,8] have a very poor descriptive capacity. In general, the curvelet coefficient distribution in natural images is characterized by a sharper peak at zero with smooth tails. This shape is associated to the sparse property of this transformation, i.e., few coefficients have high probability. This leptokurtic pattern has been previously observed in curvelets [7,12] as well as in wavelets [13]. This work proposes a texture characterization via the marginal distribution of the subband curvelet coefficients, specifically using the parameters of a generalized Gaussian density. Recent experimentation in natural images [7] shows that the generalized Gaussian density provides a good adjustment to the marginal density of the curvelet coefficient within each subband. The GGD reads as $p(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x|/\alpha)^{\beta}}$, where $\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt$, $z > 0$ is the Gamma function, α is the variance and β is related to the decreasing rate of the GGD. The parameters α and β are estimated from the subbband data through Maximum Likelihood, as is detailed in [13]. The parameters (α,β) may be used as descriptor of the probability density function of the energy levels inside each curvelet subband.

2.3 Similarity Measure

The similarity between subband curvelets is measured using the Kullback-Leibler divergence (KLD) of the corresponding GGDs:

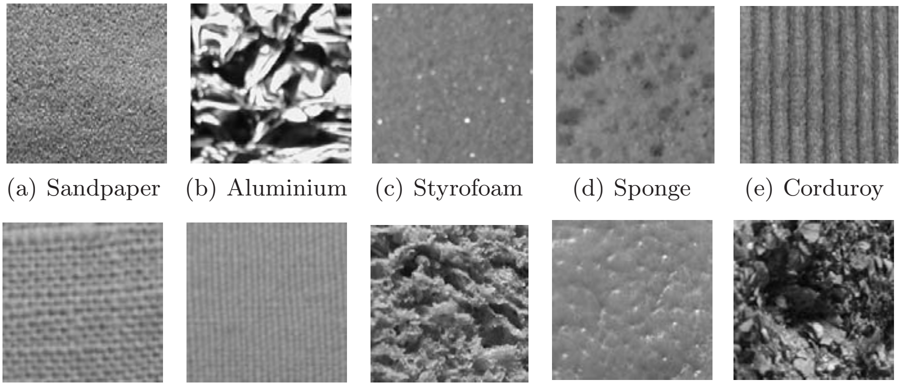
$$D(p(.; \alpha_1, \beta_1)||p(.; \alpha_2; \beta_2)) = \log\left(\frac{\beta_1 \alpha_2 \Gamma(1/\beta_2)}{\beta_2 \alpha_1 \Gamma(1/\beta_1)}\right) + \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta_2} \frac{\Gamma((\beta_2+1)/\beta_1)}{\Gamma((1/\beta_1)} - \frac{1}{\beta_1}$$

where (α_1, β_1) and (α_2, β_2) are the GGD parameters estimated for each subband. This metric does not require additional normalization and shows good performance in other multiscale domains [13]. Finally, under the reasonable assumption that curvelet coefficients in different subbands are independent, the similarity between two images I_1 and I_2 is measured as the sum of the distances between corresponding subbands $D(I_1, I_2) = \sum_{\forall s} \sum_{\forall \theta} D(p(.; \alpha_1^{s,\theta}; \beta_1^{s,\theta})||p(.; \alpha_2^{s,\theta}; \beta_2^{s,\theta}))$, where $(\alpha_1^{s,\theta}, \beta_1^{s,\theta})$ and $(\alpha_2^{s,\theta}, \beta_2^{s,\theta})$, are the GGD parameters estimated for corresponding subbands, i.e., subbands in the same scale s and orientation θ.

3 Experimental Results

The proposed descriptor was evaluated using the KTH-TIPS[1] image texture database. This database provides several variations of scale, pose and illumination and is mainly focused on classification applications; these changes increase the intra-class variability and reduce the inter-class separability, which can increase the difficulty of the classification task compared to typical databases [14]. The data consists of ten texture categories: sandpaper (*sn*), aluminium foil (*af*), styrofoam (*sf*), sponge (*sp*), corduroy (*cd*), linen (*ln*), cotton (*ct*), brown bread (*bb*), orange peel (*op*), cracker (*cr*). These real world images come from different natural scenes and have different orientations and scales. For our experiments, 45 images of each category were converted to gray-scale levels (computed from the luminance component) and cropped to 128 × 128. Figure 3 displays examples of the original textures. A real digital curvelet transform with 4 scales and 32 orientations was used, resulting in 66 subbands. The coarsest curvelet level was excluded in order to obtain robustness to changes in illumination. The algorithms are written in Matlab and run on a Intel Xeon X5460 Quad-Core 3.16 *GHz* with 8 *GB* in RAM.

The objective of the experimentation was to determine the power of description of our feature. Provided that our main goal was to assess the discriminative power of the curvelet descriptor, the feature performance in a multiclass problem was assessed using the most simple classifier, a nearest neighbour, and compared



(a) Sandpaper (b) Aluminium (c) Styrofoam (d) Sponge (e) Corduroy

(f) Linen (g) Cotton (h) Brown bread (i) Orange peel (j) Cracker

Fig. 3. Images example of several textures

[1] http://www.nada.kth.se/cvap/databases/kth-tips/index.html

Table 1. Confusion matrix for a feature based on energy and Euclidian metric

		Assigned										*Total*	*% Agree*
		sn	*af*	*sf*	*sp*	*cd*	*ln*	*ct*	*bb*	*op*	*cr*		
True	*sn*	21	0	14	7	0	0	0	2	1	0	45	0.47
	af	0	42	1	0	0	0	0	1	1	0	45	0.93
	sf	5	1	35	3	0	0	0	0	0	1	45	0.78
	dp	2	0	1	38	2	0	0	1	1	0	45	0.84
	cd	0	0	0	2	31	0	0	6	6	0	45	0.69
	ln	0	0	0	0	1	42	0	0	1	1	45	0.93
	ct	0	0	0	1	0	0	40	3	1	0	45	0.89
	bb	2	0	0	1	1	0	0	36	4	1	45	0.80
	op	0	0	1	1	2	0	0	2	27	12	45	0.60
	cr	0	0	1	0	0	0	0	1	2	41	45	0.91
	Total	30	43	53	53	37	42	40	52	44	56	450	**0.78**

Table 2. Confusion matrix for a feature based on mean, variance and Euclidian metric

		Assigned										*Total*	*% Agree*
		sn	*af*	*sf*	*sp*	*cd*	*ln*	*ct*	*bb*	*op*	*cr*		
True	*sn*	29	0	9	5	0	0	0	2	0	0	45	0.64
	af	0	41	1	0	0	0	0	1	0	2	45	0.91
	sf	4	0	38	2	0	0	0	0	0	1	45	0.84
	sp	3	0	0	39	2	0	0	0	1	0	45	0.87
	cd	0	0	0	2	31	0	0	7	5	0	45	0.69
	ln	0	0	0	0	1	44	0	0	0	0	45	0.98
	ct	1	0	0	0	2	0	41	1	0	0	45	0.91
	bb	2	0	0	2	1	0	0	35	5	0	45	0.78
	op	0	0	0	1	2	0	0	1	36	5	45	0.80
	cr	0	0	0	0	0	0	0	0	1	44	45	0.98
	Total	39	41	48	51	39	44	41	47	48	52	450	**0.84**

Table 3. Confusion matrix for our proposed feature: GGD and KLD metric

		Assigned										*Total*	*% Agree*
		sn	*af*	*sf*	*sp*	*cd*	*ln*	*ct*	*bb*	*op*	*cr*		
True	*sn*	31	0	4	5	0	0	0	5	0	0	45	0.69
	af	0	45	0	0	0	0	0	0	0	0	45	1.00
	sf	3	0	38	1	0	0	0	0	0	3	45	0.84
	sp	2	0	0	38	2	0	0	3	0	0	45	0.84
	cd	0	0	0	2	32	0	0	6	3	2	45	0.71
	ln	0	0	0	0	0	44	0	0	1	0	45	0.98
	ct	0	0	0	0	2	0	43	0	0	0	45	0.96
	bb	4	0	0	2	0	0	0	39	0	0	45	0.87
	op	1	0	1	1	0	0	0	0	42	0	45	0.93
	cr	0	1	0	0	1	0	0	1	1	41	45	0.91
	Total	41	46	43	49	37	44	43	54	47	46	450	**0.87**

with other curvelet representation methods, namely: energy of the curvelet subband plus Euclidian metric [6,8], mean and variance plus Euclidian metric [8] and the herein described proposal GGD plus KLD metric. Sumana [8] has previously compared Gabor, wavelets and curvelets, obtaining a better performance for the latter so that our work is focused on characterizing curvelets. The three classifiers were tested under a Leave-one-out cross-validation, using a single observation from the original sample as the validation data, and the remaining observations as the training data. For the three sets of experiments we computed the corresponding confusion matrix. The confusion matrices for these cases are shown in Tables 1, 2 and 3. The correct classification rates of 78%, 84% and 87% show a high discriminative capacity provided by the curvelet representation, even though we used the simpler classifier. The curvelet descriptor shows a better classification rate in both average and individually for most classes, when compared with mean and variance. Note that textures *linen (ln)* and *cotton (ct)* present a high density of lines and are correctly classified in a large number of cases. Likewise, the texture *Aluminium (al)*, which presents gross edges, is correctly classified using the curvelet descriptor. Finally, the confusion matrices show that most misclassifications occur in similar textures, for example, *sandpaper (sn)* and *styrofoam (sf)*, probably because of the similar edge distributions. In any case, the curvelet descriptor shows less classification errors even in this complicated scenario. These results show that in textures with higher levels of variability, the proposed method outperforms the previous approach. Nevertheless an extensive experimentation is needed to to draw more general conclusions.With respect to the computational complexity, the curvelet implementation runs in $O(n^2 \log(n))$ for $n \times n$ cartesian arrays [10] with a computational time that less than 300 ms for each image, while the statistical characterization for the curvelet subbands runs in less that 1 second.

4 Conclusions

We have introduced a new texture descriptor for images, based on curvelets and a statistical model of the curvelet coefficients in natural images. By applying the curvelet transform and adjusting the levels of energy for each subband to a generalized Gaussian model, we obtain a robust representation which captures the edge distribution at different orientation and scales. Experimental results indicate that the new feature improves classification performance in a multiclass problem when compared with other features, also based on curvelets. Future works includes improving the feature with invariance to rotation and scale and extensive experimentation in large texture databases.

References

1. Valerie, R.: Introduction to Texture Analysis: Macrotexture, Microtexture and Orientation Mapping. CRC Press, Amsterdam (2000)
2. Randen, T., Hå, J.: Filtering for texture classification: A comparative study. IEEE Trans. Pattern Anal. Mach. Intell. 21(4), 291–310 (1999)

3. Welland, G.: Beyond Wavelets. Academic Press, San Diego (2003)
4. Wouwer, G.V.D., Scheunders, P., Dyck, D.V.: Statistical texture characterization from discrete wavelet representations. IEEE Transactions on Image Processing 8, 592–598 (1999)
5. Kullback, S.: The kullback-leibler distance. The American Statistician (41), 340–341 (1987)
6. Dettori, L., Semler, L.: A comparison of wavelet, ridgelet, and curvelet-based texture classification algorithms in computed tomography. Comput. Biol. Med. 37(4), 486–498 (2007)
7. Alecu, A., Munteanu, A., Pizurica, A., Cornelis, W.P.J., Schelkens, P.: Information-theoretic analysis of dependencies between curvelet coefficients, pp. 1617–1620 (2006)
8. Sumana, I., Islam, M., Zhang, D., Lu, G.: Content based image retrieval using curvelet transform. In: Proc. of IEEE International Workshop on Multimedia Signal Processing MMSP, pp. 11–16 (2008)
9. Field, D.J.: Scale-invariance and self-similar 'wavelet' transforms: an analysis of natural scenes and mammalian visual systems, pp. 151–193. Elsevier-Health Sciences Division (1993)
10. Candes, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. Multiscale Modeling and Simulation 5(3), 861–899 (2006)
11. Candes, E.: New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction. Signal Processing 82(25), 1519–1543 (2002)
12. Boubchir, L., Fadili, M.J.: Multivariate statistical modeling of images with the curvelet transform. In: Eighth International Conference on Signal Processing and Its Applications - IEEE ISSPA 2005, pp. 747–750 (2005)
13. Do, M., Vetterli, M.: Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. IEEE Transactions on Image Processing 11(2), 146–158 (2002)
14. Kondra, S., Torre, V.: Texture classification using three circular filters. In: ICVGIP, pp. 429–434 (2008)

Improving Fingerprint Matching Using an Orientation-Based Minutia Descriptor

Miguel Angel Medina-Pérez, Andrés Gutiérrez-Rodríguez,
and Milton García-Borroto

Centro de Bioplantas, C. de Ávila, Cuba
{migue,andres,mil}@bioplantas.cu
http://www.bioplantas.cu

Abstract. This paper reviews a well-known fingerprint matching algorithm that uses an orientation-based minutia descriptor. It introduces a set of improvements to the algorithm that increase the accuracy and speed, using the same features. The most significant improvement is in the global minutiae matching step, reducing the number of local matching minutiae and using multiple minutiae pairs for fingerprint alignment. We conduct a series of experiments over the four databases of FVC2004, showing that the modified algorithm outperforms its predecessor and other algorithms proposed in the literature.

Keywords: biometrics, fingerprint matching, orientation-based minutia descriptor.

1 Introduction

Fingerprint recognition [1] has become one of the most active research areas nowadays. It plays an important role in forensic applications, but its increasing popularity is perhaps due to its integration into civilian systems. A key point in most of its applications is the fingerprint matching algorithm.

Most of the authors distinguish two types of fingerprint matching algorithms: correlation-based matching and minutiae-based matching. As it is seen in the Fingerprint Verification Competitions (FVC) [2], the minutia-based matching is the most popular approach. This approach essentially consists on finding the maximum number of matching minutiae pairs given two fingerprints represented by their minutiae.

Minutiae are the points where the ridge continuity breaks and it is typically represented as a triplet (x, y, θ); where x and y represents the point coordinates and θ, the ridge direction at that point. As pointed out by Feng in [3], this representation makes ambiguous the process of minutia pairing. A way to deal with this problem is enriching the minutia representation with additional information known as *minutia descriptors*. Minutia descriptors can be mainly classified in: *ridge based descriptors* [4-8], *orientation based descriptors* [4, 9-11] and *local neighboring minutiae based descriptors* [3, 12-14].

This paper reviews the algorithm created by Tico and Kuosmanen [11] (TK hereafter), and proposes improvements in the matching algorithm using the same minutia descriptor. The new proposal increases the accuracy in terms of ERR, ZeroFMR,

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 121–128, 2009.

1000FMR and 100FMR. It also reduces the matching time according to the evaluation protocols of FVC [2].

We have structured the rest of the paper as follows. Section 2 describes TK algorithm. Section 3 describes the new formulation and improvements of TK and provides some details on the implementation of the new algorithm. Section 4 details the experimental results over FVC2004. Finally, the paper provides conclusions about the new algorithm that might be the starting point for new researches.

2 TK Algorithm

TK compares fingerprints by means of local minutiae structures, also known as minutiae descriptors. Minutiae descriptors provide additional information that enriches the minutia; and they are usually invariant to rotation and translation. TK uses a descriptor based on the estimations of the orientation values on sampling points that have been arranged in concentric circles around the minutia (see Fig. 1).

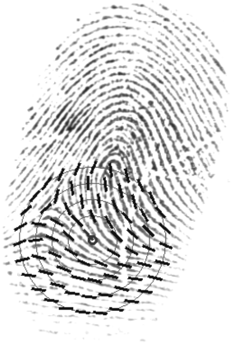


Fig. 1. A representation of the minutia descriptor proposed in [11]

Let L be the amount of circles with K_l sampling point each; given a minutia $\boldsymbol{q} = (x, y, \theta)$ we express its associated descriptor as:

$$e(\boldsymbol{q}) = \left(\left(\delta_{l,k}\right)_{k=1}^{K_l}\right)_{l=1}^{L} . \tag{1}$$

Where $\delta_{l,k}$ is the angle difference between the minutia direction $\theta \in [0,2\pi[$ and the fingerprint orientation value $\vartheta_{l,k} \in [0, \pi[$ in the k^{th} point of the l^{th} circle (the reader can refer to [1] for the conceptual difference between *direction* and *orientation*). We compute the angle difference as:

$$\delta_{l,k} = \min\{d(\theta, \vartheta_{l,k}), d(\theta, \vartheta_{l,k} + \pi)\}, \tag{2}$$

$$d(\alpha, \beta) = \min\{|\alpha - \beta|, 2\pi - |\alpha - \beta|\} . \tag{3}$$

Equation (2) computes the minimum angle required to make two lines parallel, if they have angles θ and $\vartheta_{l,k}$ respectively.

TK consist of two major steps: local minutiae matching and global minutiae matching. In the local minutiae matching step, for each query minutia $\boldsymbol{q}_j \in Q = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_n\}$ and for each template minutia $\boldsymbol{p}_i \in T = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_m\}$, the algorithm computes the *possibility value* as follows:

$$P(\boldsymbol{q}_j, \boldsymbol{p}_i) = \frac{s(\boldsymbol{q}_j,\boldsymbol{p}_i)^2}{\sum_{\substack{h=1\\h\neq j}}^{n} s(\boldsymbol{q}_h,\boldsymbol{p}_i) + \sum_{\substack{h=1\\h\neq i}}^{m} s(\boldsymbol{q}_j,\boldsymbol{p}_h) - s(\boldsymbol{q}_j,\boldsymbol{p}_i)}. \tag{4}$$

This expression returns high values when the similarity value $s(\boldsymbol{q}_j, \boldsymbol{p}_i)$ is large, and minutiae $\boldsymbol{q}_j$ and $\boldsymbol{p}_i$ have small similarities with respect to the other minutiae from $Q\backslash\{\boldsymbol{q}_j\}$ and $T\backslash\{\boldsymbol{p}_i\}$ respectively. Let $K = \sum_{l=1}^{L} K_l$, given the minutiae $\boldsymbol{q}_j$ and $\boldsymbol{p}_i$, TK computes their similarity as:

$$s(\boldsymbol{q}_j, \boldsymbol{p}_i) = 1/K \sum_{l=1}^{L} \sum_{k=1}^{K_l} \exp\left(-16(2/\pi)\left(\left|\delta_{l,k}^{j} - \delta_{l,k}^{i}\right|\right)\right). \tag{5}$$

In the global minutiae matching step, TK sort all minutiae pairs in descendent order, according to their possibility value. It transforms the query minutiae according to the minutiae pair that maximizes the possibility value. Then, it uses a greedy algorithm to find the minutiae pairs that satisfy the following constraints:

- The Euclidean distance between the two minutiae does not exceed threshold t_s.
- The difference between the two minutiae directions does not exceed threshold t_θ.

Finally, TK uses the global matching minutiae, together with those minutiae that fall inside the region of interest that is common to both fingerprints, to compute the matching score. The minutiae count inside the region of interest common to both fingerprints must exceed threshold t_m.

The parameters of the algorithm are: distance threshold t_s, angle threshold t_θ and minutia count threshold t_m.

The next section analyzes some of the drawbacks of this algorithm and proposes modifications to overcome these limitations.

3 The Modified TK Algorithm

TK algorithm only uses the local matching minutiae pair that maximizes the possibility value to align fingerprints in the global minutiae matching step. The fact that a minutiae pair maximizes the possibility value does not guarantee that this is a true matching minutiae pair (see Fig. 2). Moreover, if the selected minutiae pair was a true matching pair, it is not necessarily the best pair to carry out fingerprint alignment.

In order to deal with this limitation, the new algorithm first reduces the local matching minutiae pairs by selecting, for each query minutia $\boldsymbol{q}_j$ and template minutia $\boldsymbol{p}_i$, only the minutiae pair $(\boldsymbol{q}_j, \boldsymbol{p}_i)$ that maximizes the possibility value. Then it

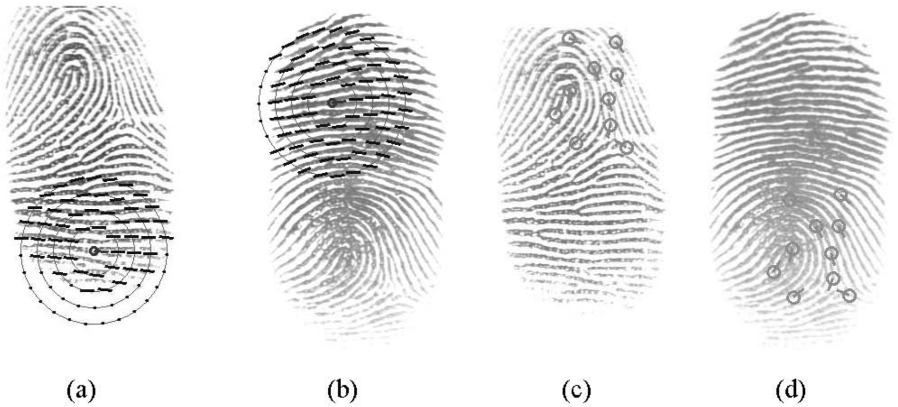


Fig. 2. The matching minutiae found by TK in fingerprints (a) 11_1 and (b) 11_3 from DB1_A in FVC2004. This is an example of a false matching minutiae pair that maximizes the possibility value in two fingerprints from the same finger. Images (c) and (d) show the good behavior of the modified TK algorithm while the original TK fails.

performs a query minutiae transformation for each minutiae pair in the reduced set. Finally, it selects the transformation that maximizes the amount of global matching minutiae pairs. This modification, like the global minutiae matching step of TK, has quadratic time complexity with respect to fingerprint minutiae count.

Another weakness of TK algorithm is that it does not take advantage of the small image rotation on the fingerprint verification problems. Therefore, we propose a modification of equation (5), which increases the minutia discrimination for such problems. We define the new minutia similarity as follows:

$$s(\boldsymbol{q}_j, \boldsymbol{p}_i) = \begin{cases} 0 & \text{if } d(\theta_j, \theta_i) > \frac{\pi}{4} \\ 1/K \sum_{l=1}^{L} \sum_{k=1}^{K_l} \exp\left(-16(2/\pi)\left(\left|\delta_{l,k}^{j} - \delta_{l,k}^{i}\right|\right)\right) & \text{otherwise} \end{cases}. \quad (6)$$

The last modification that we propose is to limit the minimum count of global matching minutiae instead of bounding the minimum minutiae count inside the region of interest that is common to both fingerprints. We introduce this modification based on the forensic criterion that a true matching fingerprints pair must have at least t_m true matching minutiae pairs [1] (t_m varies for different countries).

We name the new formulation of TK as Modified TK algorithm (MTK). A formal description of MTK is the following:

1. Let $T = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_m\}$ and $Q = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_n\}$ be the template and query fingerprint minutiae set respectively. For each query minutia $\boldsymbol{q}_j \in Q$ and for each template minutia $\boldsymbol{p}_i \in T$, compute the possibility value using equation (4).
2. Sort in descendent order all pairs $(\boldsymbol{q}_j, \boldsymbol{p}_i)$ according to the possibility value and store in $R \leftarrow \{(\boldsymbol{q}_{j,1}, \boldsymbol{p}_{i,1}), (\boldsymbol{q}_{j,2}, \boldsymbol{p}_{i,2}), \dots, (\boldsymbol{q}_{j,nm}, \boldsymbol{p}_{i,nm})\}$.
3. Set $E \leftarrow \{\}$ and $R' \leftarrow \{\}$.

```
4. For each $(\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h}) \in R$, $h = 1, \dots, nm$ do:
   a. If $\boldsymbol{q}_{j,h} \notin E \vee \boldsymbol{p}_{i,h} \notin E$ then update $R' \leftarrow R' \cup \{(\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h})\}$ and $E \leftarrow E \cup \{\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h}\}$.
5. Set $Cs \leftarrow \{\}$ and $Qs' \leftarrow \{\}$.
6. For each $(\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h}) \in R'$, $h = 1, \dots, n$ do:
   a. Set $E \leftarrow \{\}$, $C_h \leftarrow \{\}$ and $Q'_h \leftarrow \{\}$.
   b. For each $(\boldsymbol{q}_{j,g}, \boldsymbol{p}_{i,g}) \in R'$, $g = 1, \dots, n$; if $\boldsymbol{q}_{j,g} \notin E \vee \boldsymbol{p}_{i,g} \notin E$ do:
      i. Compute $\boldsymbol{q}'_{j,g} = (x'_{j,g}, y'_{j,g}, \theta'_{j,g})$ as follows:
         $\begin{bmatrix} x'_{j,g} \\ y'_{j,g} \\ \theta'_{j,g} \end{bmatrix} = \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta & 0 \\ \sin\Delta\theta & \cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{j,g} - x_{j,h} \\ y_{j,g} - y_{j,h} \\ \theta_{j,g} - \theta_{j,h} \end{bmatrix} + \begin{bmatrix} x_{i,h} \\ y_{i,h} \\ \theta_{i,h} \end{bmatrix}$ where $\boldsymbol{q}_{j,h} = (x_{j,h}, y_{j,h}, \theta_{j,h})$, $\boldsymbol{p}_{i,h} = (x_{i,h}, y_{i,h}, \theta_{i,h})$, $\Delta\theta = \theta_{i,h} - \theta_{j,h}$, $\boldsymbol{q}_{j,g} = (x_{j,g}, y_{j,g}, \theta_{j,g})$.
      ii. Update $Q'_h \leftarrow Q'_h \cup \{\boldsymbol{q}'_{j,g}\}$.
      iii. Let $\boldsymbol{p}_{i,g} = (x_{i,g}, y_{i,g}, \theta_{i,g})$; if $\sqrt[2]{(x'_{j,g} - x_{i,g})^2 + (y'_{j,g} - y_{i,g})^2} \leq t_s$ and $d(\theta'_{j,g}, \theta_{i,g}) \leq t_\theta$, then update $E \leftarrow E \cup \{\boldsymbol{q}_{j,g}, \boldsymbol{p}_{i,g}\}$ and $C_h \leftarrow C_h \cup \{(\boldsymbol{q}_{j,g}, \boldsymbol{p}_{i,g})\}$.
   c. Update $Cs \leftarrow Cs \cup \{C_h\}$ and $Qs' \leftarrow Qs' \cup \{Q'_h\}$
7. Select $C_a \in Cs$, $Q'_a \in Qs'$ where $a = \operatorname{argmax}_{h=1,\dots,n}|C_h|$.
8. If $|C_a| < t_m$ return 0 else: let $T_{Q'}$ and $Q'_T$ represent the number of minutiae from $T$ and $Q'_a$ respectively placed inside the intersection of the two fingerprint bounding rectangles, return $\frac{1}{T_{Q'}Q'_T}\left(\sum_{(\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h}) \in C_a} s(\boldsymbol{q}_{j,h}, \boldsymbol{p}_{i,h})\right)^2$.
```

Distance threshold t_s, angle threshold t_θ and minutia count threshold t_m are parameters of the algorithm. The reader can refer to Fig. 2 to see the good behavior of the modified TK algorithm in a case where the original TK fails.

4 Experimental Results

In order to evaluate the new formulation of TK algorithm we make use of the four databases of FVC2004 and the performance evaluation protocols of this competition [2]. We express in percentage the performance indicators EER, FMR100, FMR1000 and ZeroFMR. The indicator Time refers to average matching time in milliseconds.

We use the same features and parameters for both TK and MTK algorithms in all databases: distance threshold $t_s = 12$, angle threshold $t_\theta = \pi/6$ and minutia count threshold $t_m = 6$. We compute the features using the parameters that reported the best results in [11]. We carry out all the experiments on a laptop with an Intel Core Duo processor (1.86 GHz) and 1GB of RAM.

Tables 1 to 4 show the experimental results of MTK compared to: the original TK, the minutiae matching algorithm proposed by Qi et al. [9] and the results reported by Het et al. [6]. The average matching time reported in [6] does not appear in these tables because the experiments were performed using a different computer hardware. We highlight with bold letter the best result for each performance indicator.

As we expected the algorithm MTK outperform TK in all the databases for each indicator. MTK is faster than TK because the modification that we introduce in equation (6) allows discarding several false matching minutiae without comparing the respective whole descriptors. Another important result is that MTK outperforms the rest of the algorithms in most of the databases.

MTK has some limitations despite the good results achieved in the experiments. The main factors that affect the good behavior of this algorithm are high fingerprint distortion and small fingerprint area (see Fig. 3).

Table 1. Experimental results on DB1_A of FVC2004

Algorithm	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)	Time(ms)
MTK	**7.63**	**15.82**	**22.07**	**28.54**	**7.37**
TK	16.07	28.71	36.96	48.14	16.92
Qi et al. [9]	27.87	64.54	78.25	94.21	40.59
He et al. [6]	9.33	18.5	25.03	30.28	-

Table 2. Experimental results on DB2_A of FVC2004

Algorithm	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)	Time(ms)
MTK	**5.72**	**7.86**	**12.50**	**15.32**	**6.12**
TK	8.45	13.43	20.25	26.18	13.77
Qi et al. [9]	28.49	59.32	68.64	90.64	31.53
He et al. [6]	7.34	13.39	16.6	19.89	-

Table 3. Experimental results on DB3_A of FVC2004

Algorithm	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)	Time(ms)
MTK	**3.77**	**7.36**	**13.64**	**20.68**	**10.98**
TK	9.13	20.14	28.75	32.21	24.45
Qi et al. [9]	20.16	49.29	69.86	89.86	65.92
He et al. [6]	8.52	13.1	16.53	22.53	-

Table 4. Experimental results on DB4_A of FVC2004

Algorithm	EER(%)	FMR100(%)	FMR1000(%)	ZeroFMR(%)	Time(ms)
MTK	6.78	7.61	9.43	10.54	**6.48**
TK	7.72	11.21	18.07	43.25	14.18
Qi et al. [9]	26.15	60.29	70.32	82.25	33.09
He et al. [6]	**2.71**	**4.21**	**5.57**	**7.0**	-

High fingerprint distortion causes minutia descriptors with distorted orientations values while small fingerprint area causes minutia descriptors with too many sampling points with no information at all; in both cases, the minutia descriptor matching is unreliable. Minutia descriptors based on local neighboring minutiae (see [12-14]) tends to be more robust to these problems. Therefore, MTK could be improved by two ways: enriching the minutia descriptor with local neighboring minutiae information or combining MTK with an algorithm based on local neighboring minutiae descriptor.

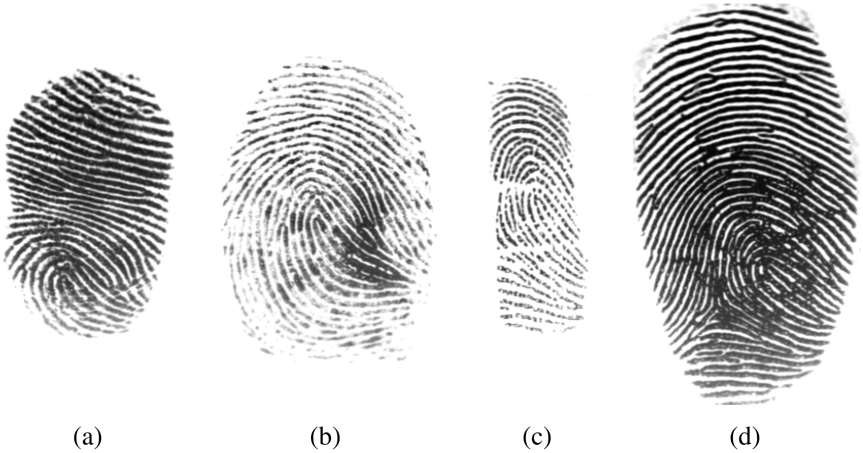


Fig. 3. These are two pairs of false not matching fingerprints from DB1_A using MTK. Fingerprints (a) 17_3 and (b) 17_6 are false not matching due to the high distortion on fingerprint (a). Fingerprints (c) 10_2 and (d) 10_4 are false not matching due to the small area of fingerprint (c).

5 Conclusions

This paper presents improvements to the fingerprint matching algorithm proposed by Tico and Kuosmanen in [11]. The new algorithm, named MTK, has three modifications of the original algorithm. First, we reduce the local matching minutiae pairs and use them all to accomplish a better fingerprint alignment. We introduce this modification because the minutiae pair that maximizes the possibility value is not necessarily a true matching minutiae pair; therefore, relying only on this pair for alignment may lead to false negative fingerprints matching. Second, we introduce a modification in the minutia similarity function in order to increase the minutiae discrimination with the additional advantage of reducing the matching time for fingerprint verification problems. Third, we include the forensic criterion that a true matching fingerprints pair must have at least certain count of true matching minutiae pairs. The conjunction of these modifications in MTK proves to be more accurate and faster than the original algorithm. The next step in our research is to investigate how the extensions of these modifications to other matching algorithms affect their performance.

References

1. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, London (2009)
2. Cappelli, R., Maio, D., Maltoni, D., Wayman, J.L., Jain, A.K.: Performance evaluation of fingerprint verification systems. IEEE Trans. Pattern Anal. Mach. Intell. 28, 3–18 (2006)
3. Feng, J.: Combining minutiae descriptors for fingerprint matching. Pattern Recognit. 41, 342–352 (2008)
4. Wang, X., Li, J., Niu, Y.: Fingerprint matching using Orientation Codes and PolyLines. Pattern Recognit. 40, 3164–3177 (2007)
5. Feng, J., Ouyang, Z., Cai, A.: Fingerprint matching using ridges. Pattern Recognit. 39, 2131–2140 (2006)
6. He, Y., Tian, J., Li, L., Chen, H., Yang, X.: Fingerprint Matching Based on Global Comprehensive Similarity. IEEE Trans. Pattern Anal. Mach. Intell. 28, 850–862 (2006)
7. Luo, X., Tian, J., Wu, Y.: A minutiae matching algorithm in fingerprint verification. In: 15th International Conference on Pattern Recognition, Barcelona, Spain, vol. 4, pp. 833–836 (2000)
8. Jain, A.K., Lin, H., Bolle, R.: On-Line Fingerprint Verification. IEEE Trans. Pattern Anal. Mach. Intell. 19, 302–314 (1997)
9. Qi, J., Yang, S., Wang, Y.: Fingerprint matching combining the global orientation field with minutia. Pattern Recognit. Lett. 26, 2424–2430 (2005)
10. Tong, X., Huang, J., Tang, X., Shi, D.: Fingerprint minutiae matching using the adjacent feature vector. Pattern Recognit. Lett. 26, 1337–1345 (2005)
11. Tico, M., Kuosmanen, P.: Fingerprint matching using an orientation-based minutia descriptor. IEEE Trans. Pattern Anal. Mach. Intell. 25, 1009–1014 (2003)
12. Feng, Y., Feng, J., Chen, X., Song, Z.: A Novel Fingerprint Matching Scheme Based on Local Structure Compatibility. In: 18th International Conference on Pattern Recognition, Hong Kong, vol. 4, pp. 374–377 (2006)
13. Jiang, X., Yau, W.Y.: Fingerprint Minutiae Matching Based on the Local and Global Structures. In: 15th International Conference on Pattern Recognition, Barcelona, Spain, vol. 2, pp. 1038–1041 (2000)
14. Ratha, N.K., Bolle, R.M., Pandit, V.D., Vaish, V.: Robust fingerprint authentication using local structural similarity. In: Fifth IEEE Workshop on Applications of Computer Vision, Palm Springs, CA, USA, pp. 29–34 (2000)

Morphological Shape Context: Semi-locality and Robust Matching in Shape Recognition

Mariano Tepper[1], Francisco Gómez[1], Pablo Musé[2], Andrés Almansa[3], and Marta Mejail[1]

[1] Universidad de Buenos Aires, Argentina
[2] Universidad de la República, Uruguay
[3] Telecom ParisTech, France

Abstract. We present a novel shape recognition method based on an algorithm to detect contrasted level lines for extraction, on Shape Context for encoding and on an *a contrario* approach for matching. The contributions naturally lead to a semi-local Shape Context. Results show that this method is able to work in contexts where Shape Context cannot, such as content-based video retrieval.

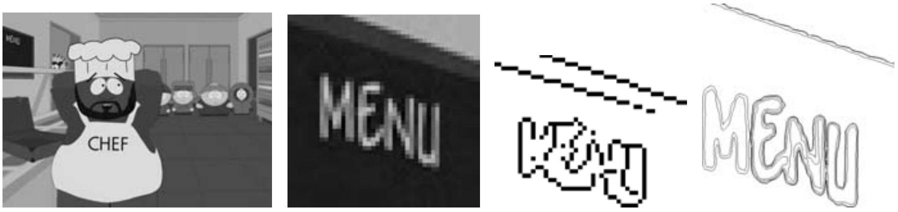
1 Introduction

The problem of Shape Matching and Recognition can be described as a three-stage process [1]: *(i)* edge detection; *(ii)* invariant coding and matching of semi-local shapes; and *(iii)* grouping of local shape matches.

The first step is most commonly solved by the use of a Canny edge detector which has at least two drawbacks: *(a)* several parameters have to be manually tuned depending on contrast and noise; and *(b)* edges are represented as a non-structured set of edge points which needs to be later grouped into curves, which is a non trivial and error prone task. In this work we substitute the Canny edge detector by a refinement of the Meaningful Boundaries (MB) algorithm [2]. The representation of edges as well-contrasted pieces of level-lines (inspired from mathematical morphology) avoids the edgel linking stage, and the use of Gestalt-inspired [3] *a contrario* detection theory [2] provides a theoretically sound and effective means of selecting parameters and the contrast/noise trade-off. In addition our refinement (see section 3) eliminates the main shortcomming of the basic MB algorithm, thus avoiding that low-contrast parts of the level-lines keep well-contrasted parts from being detected. Fig. 1 compares our MB refinement with the Canny edge detector. Observe that the use of continuous level-lines extracted from a bilinearly interpolated image provide much more finer-grained information, solve the edge-linking problem more effectively and does not introduce a significant computational penalty (thanks to the bilinear FLST [1]).

Once shapes have been extracted from the image, a suitable representation to describe them has to be chosen (step *(ii)* above). Belongie et al. proposed a shape descriptor that is called Shape Context (SC) [4]. SC has many advantages and has been used succesfully in several applications. SC encodes shapes from

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 129–136, 2009.



(a) (b) (c) (d)

Fig. 1. (a) original image; (b) an area on its upper left corner; (c) detailed view of Canny's filter applied to (a); (d) detailed view of MB applied to (a)

the edge map of an image and it therefore inherits its aforementioned drawbacks. The novel contribution of this work (see section 3) is to fuse SC and MB in what we call Morphological Shape Context (MSC). Results presented further show that this descriptor is able to work in contexts where SC cannot.

The matching step is the least studied of all the processes involved in visual recognition. Most methods use a nearest neighbor approach to match two sets of descriptors [5]. In this work we present an a contrario shape context matching criterion (see [6] and section 4), which gives a clear-cut answer to this issue.

Shape matching as described so far (step *(ii)*) only allows to match relatively simple semi-local shapes. More complex shapes will be represented by groups of shapes that are geometrically arranged in the same manner in both images. Such groups can be detected as a third clustering step. In this work we do not describe this stage in detail but use a basic RANSAC [7] implementation in section 5, in order to experimentally evaluate the results of steps *(i)* and *(ii)* in the context of content-based video retrieval applications.

2 Shape Extraction

This section addresses the problem of extracting the shapes present in an image. We make use of the Fast Level Set Transform (FLST) method where the level sets are extracted from an image, and we propose an extension of the MB algorithm [2], that detects contrasted level lines in grey level images. Let C be a level line of the image u and $x_0, x_1, \dots, x_{n-1}$ denote n regularly sampled points of C, with geodesic distance two pixels, which in the *a contrario* noise model are assumed to be independent. In particular the gradients at these points are independent random variables. For $x_i \in C$, let μ_j $(0 \leq j \leq n-1)$ be the j-th value of the increasingly sorted vector of the contrast at x_i defined by $|Du|(x_i)$ (the image gradient norm $|Du|$ can be computed on a 2×2 neighborhood).

The curve detection algorithm consists in adequately rejecting the null hypothesis $\mathcal{H}_0$*: the values of* $|Du|$ *are i.i.d., extracted from a noise image with the same gradient histogram as the image* u *itself.*

Following [8], for a given curve, the probability under $\mathcal{H}_0$ that at least k among the n values μ_j are greater than μ is given by the tail of the binomial law

$\mathcal{B}(n, k, H_c(\mu))$, where $H_c(\mu) = P(|Du| > \mu)$. The regularized beta function I can be regarded as an interpolation of the binomial tail to the continuous domain and can be computed much faster. Thus it is interesting, and more convenient, to extend this model to the continuous case using the regularized incomplete beta function $I(H_c(\mu); l_1(k), l_2(k))$, where $l_1(k) = \frac{l}{2}\frac{n-k}{n}$ and $l_2(k) = 1 + \frac{l}{2}\frac{k}{n}$. This represents the probability under $\mathcal{H}_0$ that, for a curve of length l, some parts with total length greater or equal than $l_1(k)$ have a contrast greater than μ.

Definition 1. *Let $\mathcal{C}$ be a finite set of N_{ll} level lines of u. A level line $C \in \mathcal{C}$ is an ε-meaningful boundary if $NFA(C) \equiv N_{ll} \cdot K \cdot \min_{0 \leq k < K} I(H_c(\mu_k); l_1(k), l_2(k)) < \varepsilon$, where K is a parameter of the algorithm. This number is called number of false alarms (NFA) of C.*

As in [1], the expected number of ε-meaningful boundaries in a finite random set of random curves can be proven to be smaller than ε.

Meaningful boundaries usually appear in parallel and redundant groups, because of interpolation. The shape extraction algorithm only detects curves with minimal NFA in such groups [1].

The refinement proposed in Def. 1 is no other than a relaxation of the classic definition by Desolneux *et al.* ([2]) which aims at avoiding underdetection by allowing some parts (up to $k < K$ out of n points) of the curve to be low-contrasted.

The choice of the value of K cannot be directly made as it is highly dependent on the length and the constrast of the curve. Thus the value of K has to be chosen as a function of the curve length and of the image contrast along the curve.

Following Def. 1, we set the value of K as $\hat{K}_\varphi \equiv \arg\max_{i<n} \left(\frac{\sum_{j=0}^{i} \mu_j}{\sum_{j=0}^{n-1} \mu_j} < \varphi \right)$ where $\varphi \in [0, 1]$ is the new parameter of the detection algorithm.

This choice of K is indeed adaptive to the length and contrast of each level line. It is in fact quite stable for values of $\varphi < 0.05$. Larger values lead to an overdetection and, in general, no perceptually significant level lines appear. Studying how this relates with the laws of visual perception is an interesting subject for future research. From a computational and pragmatic point of view, we consider here that this is not a critical parameter that has to be set by the user because: *(i)* all experiments were performed with the same value of $\varphi = 0.02$ obtaining near-optimal performance; and *(ii)* varying the value of φ within the range $(0, 0.05)$ does not significantly affect the results.

3 Shape Encoding

In this section we overview the SC technique [4], and we present an improved version that leads to an intrinsic definition of semi-locality in this new descriptor.

The SC considers a sampled version of the image edge map as the shape to be encoded. The SC of a point in the shape is a coarse histogram of the relative positions of the remaining points. The histogram bins are taken uniformly in log-polar space, making the descriptor more sensitive to positions of nearby sample points than to those farther away.

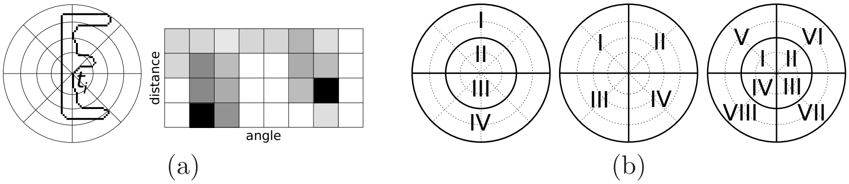


Fig. 2. (a) Shape context of a character 'E'. Left, partition into bins around the point t_i; right, matrix representation of SC_{t_i} (darker means more weigth). (b) Different ways to split a shape context. Doted lines separate bins and thick lines separate bin groupings.

Let $\mathcal{T} = \{t_1, \ldots, t_n\}$ be the set of points sampled from the edge map of an input image. For each $t_i \in \mathcal{T}$, $1 \leq i \leq n$, the distribution of the $n - 1$ remaining points in $\mathcal{T}$ is modeled relative to t_i as a log-polar histogram (Fig. 2a). We denote by $\Theta \times \Delta$ a partition of the log-polar space $[0, 2\pi] \times (0, L]$ into A and B bins respectively, where $L = \max_{t_j \in \mathcal{T}} ||t_j - t_i||_2$. The histogram is defined as

$$SC_{t_i}(\Theta_k, \Delta_m) = \#\{t_j \in \mathcal{T} \ : \ j \neq i, t_j - t_i \in (\Theta_k, \Delta_m)\}$$

where $0 < k \leq A$ and $0 < m \leq B$. The Shape Context of t_i (SC_{t_i}) is defined as a normalized version of $SC_{t_i}(\Theta_k, \Delta_m)$.

Fig. 2a depicts both spatial and matrix representations of a shape context.

The collection of the SC for every point in the shape is a redundant and powerful descriptor for that shape but has some drawbacks.

First, the sampling stage is performed by considering that the edge map corresponds to a Poisson process [4]. This hard-core model produces a non-deterministic sampling algorithm which means that different runs of the sampling algorithm may give slightly different results. The immediate consequence is that two descriptors from exactly the same image, obtained at different times, may not be equal. In short terms, jitter noise is introduced in the descriptor. In Fig. 3 the effect of the jitter noise is shown, making $d(SC_{t_i}, SC_{t_j}) \approx 0.11 \neq 0$[1].

Second, from our point of view the main drawback of SC is that it inherits the weaknesses from the edge map. We mentioned previously that extracting curves from the edge map is a hard problem. This fact has a great impact in shape encoding: there is no intrinsic distinction between what is global and what is not. An example is shown in Fig. 3, where $d(SC_{t_i}, SC_{t_k}) \approx 0.3$ which is clearly above the jitter noise $d(SC_{t_i}, SC_{t_j})$. In short terms, a slight modification of the shape has a great impact on the distance. The question "Where does a shape begin and where does it end?" becomes absolutely non trivial. The efforts to overcome this issue lead to heuristic solutions.

As stated above, the topographic map provides a natural solution to these issues. Meaningful boundaries are much more suitable than the edge map for shape recognition. Meaningful boundaries are used as the set of shapes to be encoded and recognized from an image [1]. Maximal Stable Extremal Regions

[1] $d(\cdot, \cdot)$ is the χ^2 distance and is used throughout this paper.

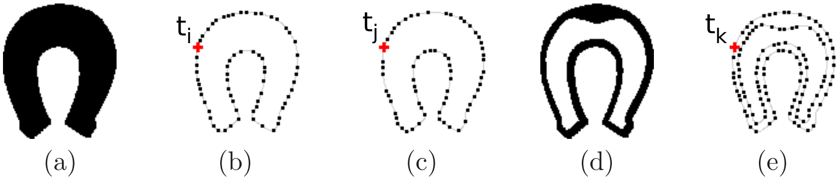


Fig. 3. (a) image horsehoe1; (b) sampled points from horsehoe1; (c) other sampled points from horsehoe1, with the same sampling process than those in (b); (d) image horsehoe2; (e) sampled points from horseshoe2 with the same sampling process than those in (b) and (c). The points t_i, t_j ad t_k are in the same position of the image.

(MSER), which are very close in spirit to MB, have also been used for shape encoding, see [9] among others.

The main idea is to exploit the benefits of the image structure representation defined in the previous section and to fuse it with SC. We call this new descriptor Morphological Shape Context (MSC).

As in SC, each shape in a given image is composed by a set of points. In MSC, we consider each curve (i.e. meaningful boundary) as a shape. When dealing with curves, the sampling stage is done in a very natural way, by arc-length parameterisation, thus eliminating jitter noise. In the resulting algorithm, shapes are extracted using the MB algorithm. Let us redefine $\mathcal{T} = \{t_1, ..., t_n\}$ as the set of points sampled from a meaningful boundary of an image. The SC is then computed for each sample point t_i, $1 \leq i \leq n$.

Beside the advantages of the representation we described above, one of its keys is the natural separation between level lines (they do not intersect). It allows to go from a global shape encoding to a semi-global one in a natural way, i.e. without fixing any arbitrary threshold. The most powerful advantage is that individual objects present in the image can be matched separately, which was not posible in SC.

In [1] the Level Line Descriptor was designed to detect that two images share *exactly* the same shape. The "perceptual invariance" is only introduced in the matching stage. That is not what we are aiming for. We want to keep the intrinsic "perceptual invariance" given by the SC and be able to detect that two images share two *similar* shapes, independently of the matching algorithm.

4 Shape Matching

As shown in [1], the *a contrario* framework is specially well suited for shape matching. Let $\mathcal{F} = \{F^k | 1 \leq k \leq M\}$ be a database of M shapes. For each shape $F^k \in \mathcal{F}$ we have a set $\mathcal{T}^k = \{t_j^k | 1 \leq j \leq n_k\}$ where n_k is the number of points in the shape. Let $SC_{t_j^k}$ be the shape context of t_j^k, $1 \leq j \leq n_k$, $1 \leq k \leq M$. As in [6] we assume that each shape context is split in C independent features that we denote $SC_{t_j^k}^{(i)}$ with $1 \leq i \leq C$ (see Fig. 2b for an example).

Let Q be a query shape and q a point of Q. We define $d_j^{k(i)} = d(SC_q^{(i)}, SC_{t_j^k}^{(i)})$. The matching algorithm consists in adequatley rejecting the null hypothesis $\mathcal{H}_0$: *the distances* $d_j^{k(i)}$ *are realizations of* C *independent random variables* $D^{(i)}$, $1 \leq i \leq C$.

Definition 2. *The pair* (q, t_j^k) *is an* ε*-meaningful match in the database* $\mathcal{F}$ *if*

$$\mathrm{NFA}(q, t_j^k) \equiv \left(\sum_{k'=1}^{M} n_{k'} \right) \cdot \prod_{i=1}^{C} P(D^{(i)} \leq d_j^{k(i)} \mid \mathcal{H}_0) < \varepsilon .$$

This number is called number of false alarms (NFA) of the pair (q, t_j^k).

This provides a simple rule to decide whether a single pair (q, t_j^k) does match or not. From one side, this is a clear advantage over other matching methods since we have an individualized assessment for the quality of each possible match. From the other side, the threshold is taken on the probability instead of directly on the distances. Setting a threshold directly on the distances d_j^k (or $d_j^{k(i)}$ for the case) is hard, since distances do not have an absolute meaning. If all the shapes in the database look alike, the threshold should be very restrictive. If they differ significantly from each other, a relaxed threshold would suffice.

Thresholding on the probability is more robust and stable. More stable, since the same threshold is suitable for different database configurations. More robust, since we explicitly control false detections. As proven in [1], the expected number of ε-meaningful matches in a random set of random matches is smaller than ε.

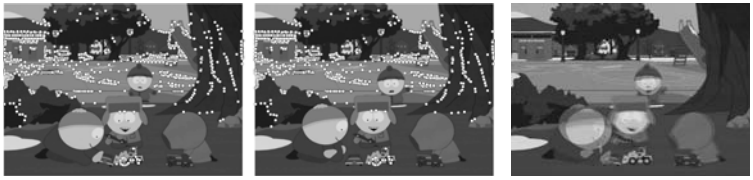
5 Results and Conclusions

In this section we illustrate the performance of the presented methods with three different examples. All the experiments in this paper were produced using $\varphi = 0.02$ for the computation of MB. In both *a contrario* algorithms taking $\varepsilon = 1$ should suffice but we set $\varepsilon = 10^{-10}$ for MB and $\varepsilon = 10^{-2}$ for matching to show the degree of confidence achievable whithout affecting the results.

In the first example, we tested the approach in a video sequence from South Park, which is textureless and composed only by contours. In Fig. 4a, meaningful matches between two consecutive frames are depicted. White dots represent the centers of the MSC. In Fig. 4b, both frames are overlapped to show moving shapes. Note that in Fig. 4a there are no matches in these areas.

The second example, displayed in Fig. 5, is closely related to the first one. Here texture is present and a non-rigid character is moving on the foreground. The matches between frames 3 and 4 of the sequence are shown. Only shapes not occluded by the movement are matched. The channel logo is correctly matched since it is located in the foreground and it does not move.

Finally, in Fig. 6 an application to content-based video retrieval is shown. We searched for the parental guidance logo in a video sequence with more than 6000 frames. Fig. 6a depicts the number of matches for each frame of the video. The



(a) (b)

Fig. 4. (a) Matches (white dots) between two frames. There are 1525 matches coherent with a similarity transformation. (b) Both frames overlapped to show moving shapes.

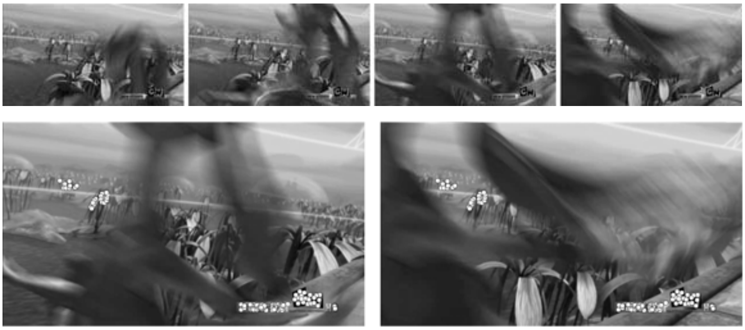
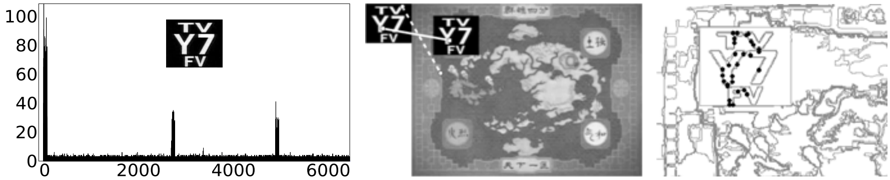


Fig. 5. A video sequence with a non-rigid character moving on the foreground (top). The channel logo is in the bottom right. Matching between frames 6 and 7: there are 141 meaningful matches (white dots) coherent with a similarity transformation.



(a) (b) (c)

Fig. 6. (a) Number of matches per frame of a video with the displayed query. (b) Best (solid line) and worst (dashed line) matches for a target frame. (c) Detail of the logo area with matched points in black dots.

logo is present in three intervals ([0, 76], [2694, 2772] and [4891, 4969]) which coincide with the three spikes. These spikes are clearly higher than spurious matches in the rest of the video. The second and third spike are smaller than the first one, in those intervals the logo is only at 66% of its original size. This is achieved without any multiscale processing. In Fig. 6b the best match (the correct one) has a NFA of $2.45 \cdot 10^{-9}$ and the worst one (the wrong one), of $9.99 \cdot 10^{-3}$. At $\varepsilon = 10^{-4}$ all matches are correct.

The same experiment as in Fig. 6b using SC gives 3 matches instead of the 29 obtained using MSC (Fig. 6c). All MSC matches are correct and all SC matches are wrong: the global SC approach is unable to match semi-local shapes.

The examples show that semi-locality in the MSC is a key feature to match shapes in contexts where other shapes are present: when very similar images present little differences (Fig. 4), when different foregrounds occlude the same background (Fig. 5), when the query is not present or surrounded by a large set of shapes (Fig. 6). MSC provides a novel approach to deal with such contexts, proving itself successful where SC is not.

References

1. Cao, F., Lisani, J.L., Morel, J.M., Musé, P., Sur, F.: A Theory of Shape Identification. Lecture Notes in Mathematics, vol. 1948. Springer, Heidelberg (2008)
2. Desolneux, A., Moisan, L., Morel, J.M.: From Gestalt Theory to Image Analysis, vol. 34. Springer, Heidelberg (2008)
3. Kanizsa, G.: Organization in Vision: Essays on Gestalt Perception. Praeger, Westport CT (1979)
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI 24(4), 509–522 (2002)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints (2003)
6. Tepper, M., Acevedo, D., Goussies, N., Jacobo, J., Mejail, M.: A decision step for shape context matching. IEEE ICIP (in press, 2009)
7. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)
8. Meinhardt, E., Zacur, E., Frangi, A., Caselles, V.: 3d edge detection by selection of level surface patches. Journal of Mathematical Imaging and Vision
9. Obdrzálek, S., Matas, J.: Object recognition using local affine frames on distinguished regions. In: Rosin, P.L., Marshall, D.A. (eds.) BMVC (2002)

On the Computation of the Common Labelling of a Set of Attributed Graphs

Albert Solé-Ribalta and Francesc Serratosa

Department of Computer Science and Mathematics
Universitat Rovira i Virgili (URV). Avda. Països Catalans, 26. 43007 Tarragona
albert.sole@urv.cat, francesc.serratosa@urv.cat

Abstract. In some methodologies, it is needed a consistent common labelling between the vertices of a set of graphs, for instance, to compute a representative of a set of graphs. This is a NP-problem with an exponential computational cost depending on the number of nodes and the number of graphs. The aim of this paper is twofold. On one hand, we aim to establish a technical methodology to define this problem for the present and further research. On the other hand, we present two sub-optimal algorithms to compute the labelling between a set of graphs. Results show that our new algorithms are able to find a consistent common labelling while reducing, most of the times, the mean distance of the AG set.

Keywords: Multiple graph matching, common graph labelling, inconsistent labelling, softassign.

1 Introduction

In some patter recognition applications, it is useful to define a representative of a set or cluster of elements. Some well-known techniques have been described when the elements are characterised by a feature vector. Nevertheless, only few techniques have been developed when the elements are represented by Attributed Graphs (AGs). When this is the case and when we want to synthesise the representative, examples could be found in: [1], [2], [3], [4] and [7], considering all the set at a time, it is needed a common labelling between each AG vertex (and arcs) and the vertices (and arcs) of the representative. Thus, given a priori this labelling, the new representative can be synthesised. Moreover, if we want the new structure to represent the cluster, it is desired that this structure is defined such that the sum of distances between the AGs and this new prototype is minimum. When this occurs, we say that we have obtained an Optimal Common Labelling of a set of AGs (OCL).

The main impediment on solving the OCL problem is that it is an NP-problem and therefore, the computational cost is exponential on the number of nodes and also on the number of AGs. For this reason, it is crucial to find algorithms that compute good approximations of the OCL in polynomial time. The aim of this paper is to present two new sub-optimal algorithms to compute the OCL of a set of AGs.

Similar works on this issue could be found in [8] where a labeling between two AG is induced by all the labelings of the set, however no OCL can be found using this

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 137–144, 2009.

procedure. Some other works, where the aim of the article is finding the OCL, will we be introduced later on the document.

The document is structured as follows. In the next section, we introduce basic concepts and notation related to AG matching. In section 3, we define and explain, in detail, the problem we want to solve. In section 4, the new algorithms are briefly introduced. Section 5 presents results and evaluation of the algorithms. Finally, section 6 summarizes the article with some conclusions and further work.

2 Definitions

Definition 1. Attributed Graph: Let Δ_v and Δ_e denote the domains of possible values for attributed vertices and arcs, respectively. An attributed graph *AG* over *(Δ_v and Δ_e)* is defined by a tuple $AG=(\Sigma_v, \Sigma_e, \gamma_v, \gamma_e)$, where $\Sigma_v=\{v_k \mid k = 1,\dots,R\}$ is the set of vertices (or nodes), $\Sigma_e=\{e_{ij} \mid i,j \in \{1,\dots,R\}, i \neq j\}$ is the set of arcs (or edges) and $\gamma_v: \Sigma_v \rightarrow \Delta_v$, $\gamma_e: \Sigma_e \rightarrow \Delta_e$ assign attribute values to vertices and arcs respectively.

Definition 2. Isomorphism between AGs: Let $G^p = (\Sigma_v^p, \Sigma_e^p, \gamma_v^p, \gamma_e^p)$ and $G^q=(\Sigma_v^q, \Sigma_e^q, \gamma_v^q, \gamma_e^q)$ be two AGs. Moreover, let T be a set of isomorphisms between two vertex sets Σ_v. The isomorphism $f^{pq}:\Sigma_v^p \rightarrow \Sigma_v^q$, $f^{pq} \in \mathrm{T}$, assigns each vertex from G^p to only one vertex of G^q. There is no need to define the arcs isomorphism since they are mapped accordingly to the node isomorphism of their terminal nodes.

Definition 3. Cost and Distance between AGs: Let f^{pq} be the isomorphism $f^{pq}:\Sigma_v^p \rightarrow \Sigma_v^q$ that assigns each vertex from G^p to a vertex of G^q. The cost of this isomorphism, $C(G^p,G^q, f^{pq})$ is a function that represents how similar are the AGs and how correct is the isomorphism. Usually, $C=0$ represents that both AGs are identical and that the isomorphism captures this similarity. The distance D between two AGs, is defined to be the minimum cost of all possible isomorphisms f^{pq}. That is, $D\left(G^p,G^q\right)=\min\limits_{f^{pq}\in Y} C\left(G^p,G^q,f^{pq}\right)$ [9]. We say that the isomorphism f^{pq} is *optimal* if it is the one used to compute the distance.

3 Common Labelling of a Set of AGs

The first step of the algorithms presented in the literature [1], [2], [3], [4] is to obtain all possible isomorphisms between all AGs of the set. Once these isomorphisms are obtained, then the Common Labelling is computed.

Definition 4. Multiple Isomorphism of a set of AGs: Let *S* be a set of *N* AGs, $S=\{G^1, G^2, \dots, G^N\}$. We say that the set *F* is a Multiple Isomorphism of *S* if *F* contains one and only one isomorphism between the AGs in *S*, $F = \{f^{1,2}, \dots, f^{2,1}, \dots, f^{N,N}\}$.

We assume that the AGs have *R* nodes. If it is not the case, the AGs would have to be extended with null nodes. We say that a multiple isomorphism is *consistent* if concatenating all the isomorphisms, we can define disjoint *partitions* of vertices. Every *partition* is supposed to contain one and only one vertex per each AG and, in

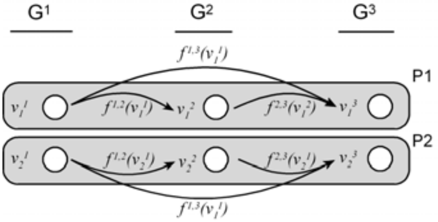


Fig. 1. Consistent multiple isomorphism

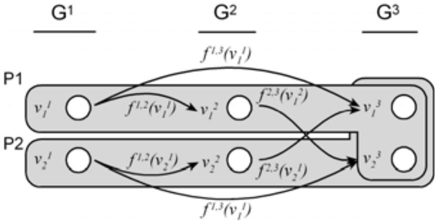


Fig. 2. Inconsistent multiple isomorphism

addition, every vertex must belong to only one partition. **Figure 1** shows a *Consistent Multiple Isomorphism* between three AGs, being *R=2*. We can distinguish two partitions, *P1* and *P2*. **Figure 2** shows the same AGs with an *Inconsistent Multiple Isomorphism*, where partitions share two nodes, hence partitions are not disjoint.

Definition 5. Consistent Multiple Isomorphism of a set of AGs (CMI): Let *F* be a Multiple Isomorphism of *S*. *F* is a CMI of *S* if it fulfils that $f^{qk}\left(f^{pq}\left(v_i^{pq}\right)\right)= f^{pk}\left(v_i^{pk}\right)$, $0< p,q,k \le N, 0< i \le R$.

Given an isomorphism, we can define its costs (**definition 3**). Extending this definition, given a CMI of a set, we define its cost as the addition of the costs of all isomorphisms. The Optimal Consistent Multiple Isomorphism (OCMI) is the CMI with the minimum cost. Note that, the cost of the OCMI may be obtained by non-optimal isomorphisms since it is restricted to be consistent.

Definition 6. Optimal Consistent Multiple Isomorphism of a set of AGs (OCMI): Let *F* be a CMI of *S*. *F* is an Optimal Consistent Multiple Isomorphism (OCMI) of *S* if it fulfils that $F = \underset{f^{pq}\in \Upsilon}{\arg\min} \sum_{\forall G^p,G^q} C\left(G^p,G^q,f^{pq}\right)$.

Given a CMI, we can define a Common Labelling (CL) of a set of AGs. Note that all the vertices of each partition are labelled to the same node of the virtual structure. For this reason, it is needed the MI to be consistent. If not, the CL would not be a function since an AG node would have to be labelled to several nodes of the virtual structure.

Definition 7. Common Labelling of a set of AGs (CL): Let *F* be a CMI of *S* and let L_v be a vertex set, $L_v \in \Sigma_v$. The Common Labelling *H=* $\{ h^1, h^2, \dots , h^n\}$ is defined to be a set of bijective mappings from the vertices of AGs to L_v as follows: $h^1(v_i^1)=i$ and $h^p(v_i^p)=h^{p-1}(v_j^{p-1})$, $1\le i,j \le R$, $2\le p\le N$, being $f^{p-1,p}(v_j^{p-1})=v_i^p$. **Figure 3** illustrates this definition.

Finally, the Optimal Common Labelling of a set is a CL computed through an OCMI. The prototype or representative of the set synthesised using this CL would be the best representative, from the statistical point of view, since the sum of the costs of each pair of AGs, considering the global consistency requirement, is the lowest among all possible CL.

Definition 8. Optimal Common Labelling of a set of AGs (OCL): Let *H* be a CL of *S* computed by a CMI *F*. We say that *H* is an Optimal Common Labelling (OCL) of *S* if *F* is an OCMI of *S*.

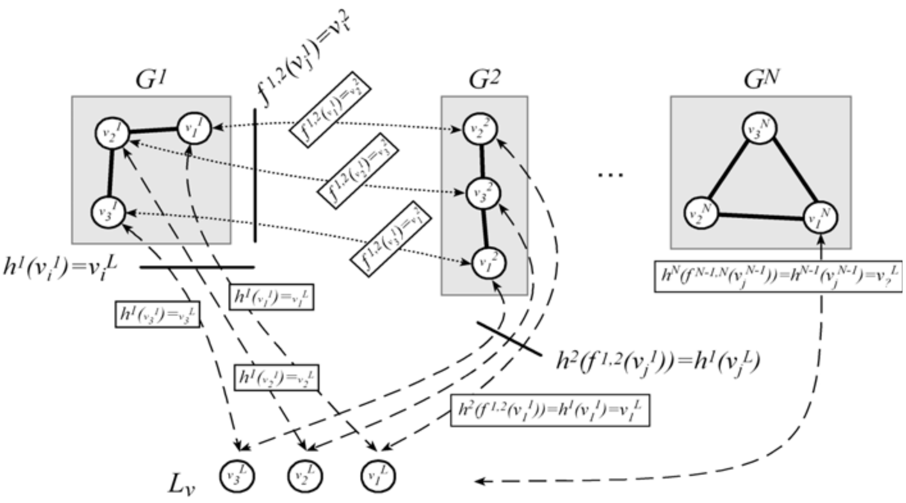


Fig. 3. Illustration of a CL given a CMI

Consistency Index

Through **definition 5**, we can discern whether a MI is consistent or not. Nevertheless, we are interested in establishing a consistency measure of a non-consistent MI to know the goodness of a concrete labelling given by sub-optimal labeling algorithms.

The *consistency index* that we propose shows the correctness of a MI taking values in the domain [0,1]. The higher values obtained the better consistency in the MI. Only in the case that the MI is consistent, the consistency index equals 1. To obtain a smooth index, we base our index on the number of inconsistencies given any triplet of AGs from the set. Thus, given the triplet of AGs, G^1, G^2 and G^3 and the MI $F = \{f^{1,2}, f^{1,3}, f^{2,3}\}$, we say that there is a tripled inconsistency if $f^{1,3}(v_i^1) \neq f^{2,3}(f^{1,2}(v_i^1))$, $1 \leq i \leq R$ (extracted from **definition 5**). The cost of this operation is linear respect the number of nodes. Finally, the *Consistency Index* is obtained as a function depending on the number of triplet inconsistencies and the number of possible triplets.

$$Consistency\,Index = 1 - |Tripled\ Inconsistency| \Big/ \binom{N}{3}$$

4 Computing the OCL

Figure 4 presents the main methodology used up to now to compute a sub-optimal solution for the Common Labelling problem[1][2][3][4]. It is composed by three main steps. First, for each pair of AGs, an assignation matrix is computed. To do so, several error-tolerant graph matching algorithms have been presented, such as, probabilistic relaxation [11], softassign [5] or Expectation-Maximisation [12]. Each cell of the assignation matrix M_{ai} stores the probability of node *a,* from G^1, to be assigned to node *i*, from graph G^2, that is, the probability of the labelling $f^{1,2}(v_i^1) = v_a^2$, $p(v_i^1, v_a^2)$.

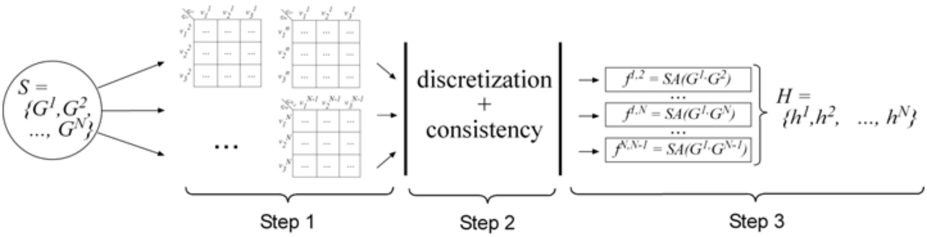


Fig. 4. Basic scheme to compute a CL based on a set of assignation matrices

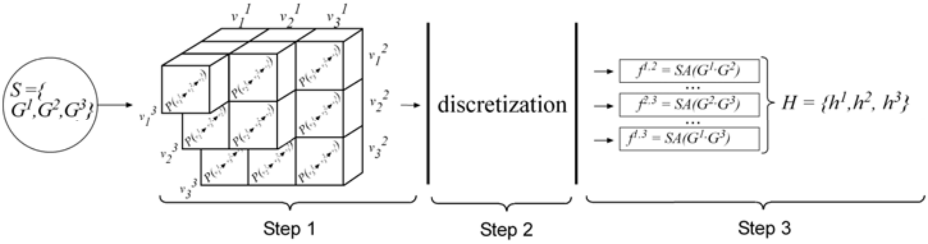


Fig. 5. New scheme to compute a CL based on an assignation hypercube. Example using N=3 graphs.

The cost to obtain this matrix, using softassign [5], is $O((N/2\text{-}N)\cdot R^4)$ per iteration of the algorithm.

In the second step, some times called *clean-up* process, the CMI of the set is obtained. Again, several techniques can be used, but, all of them consider the probabilities of all the individual assignation matrices and the restrictions imposed by the consistency requirements to obtain the final CMI. The cost of this step, again using softassign, is $O((N/2\text{-}N)^R)$. Finally, in the last and simplest step, the CL is defined.

Figure 5 presents our new methodology. The main difference appears in the first and second step. In the first step, the set of assignation matrices is substituted by an assignation hypercube to alleviate the problem of taking the individual assignation matrices independently. As a consequence, the first step generates always consistent isomorphisms. Therefore, the second step does not need to figure out a consistent MI. The third step is equivalent to the previous methodologies. The number of dimensions of the hypercube is the number of AGs of the set, *N*. Each cell of the hypercube $M_{a_1 a_2 \ldots a_N}$ represents the joint probability of:

$$f^{1,2}\left(v_{a_1}^1\right)=v_{a_2}^2 \;\&\; f^{1,3}\left(v_{a_1}^1\right)=v_{a_3}^3 \;\&\ldots\&\; f^{1,N}\left(v_{a_1}^1\right)=v_{a_N}^N \;\&\ldots\&\; f^{2,3}\left(v_{a_2}^2\right)=v_{a_3}^3 \;\&\ldots\&\; f^{2,N}\left(v_{a_2}^2\right)=v_{a_N}^N \ldots$$

That is, $p\left(v_{a_1}^1, v_{a_2}^2, \ldots, v_{a_N}^N\right)$.

Step 1 of the new algorithm: computing an assignation hyper-cube

To compute the assignation hyper-cube, we have developed two algorithms. In the first one, called *N-dimensional softassign*, the joint probability $p\left(v_{a_1}^1, v_{a_2}^2, \ldots, v_{a_N}^N\right)$ has to be computed all at once since the marginal probabilities are not considered independent. The algorithm is a generalisation of the softassign algorithm, in which, the double-stochastic [6] matrix is converted to an N-stochastic hyper-cube and all other parts of the algorithm are extended accordingly. It has the advantage that the whole MI is considered at once in each iteration; and therefore, the information of the partial labelings is used globally. The computational cost is *O(R^{2N})* at each iteration.

The second algorithm, called *agglomerative softassign*, is based on the supposition that the joint probability $p\left(v^1_{a_1}, v^2_{a_2}, \ldots, v^N_{a_N}\right)$ can be obtained as the product of the marginal ones since they are independent:

$$p\left(v^1_{a_1}, v^2_{a_2}, \ldots, v^N_{a_N}\right) = p\left(v^1_{a_1}, v^2_{a_2}\right) \times p\left(v^1_{a_1}, v^3_{a_3}\right) .. p\left(v^1_{a_1}, v^N_{a_N}\right) \times p\left(v^2_{a_2}, v^3_{a_3}\right) .. p\left(v^2_{a_2}, v^N_{a_N}\right) ..$$

The algorithm is composed by two main steps. In the first one, all the individual assignation matrices are computed using any of the algorithms mentioned before. Using those individual assignation matrices, in the second step, the cells of the hypercube are obtained as the product of the corresponding cells of the assignation matrices.

$$M_{v^{G_1}_{a_1}, v^{G_2}_{a_2}, \ldots, v^{G_N}_{a_N}} = \prod_{1 \le i < j \le N} M^{Gi,Gj}_{a_i,a_j}$$

The cost is the sum of computing all individual assignation matrices plus the cost of joining all those individual matrices to the hypercube *O(((N^2/2)-N)·R^N)*.

In this paper we base our extensions of “Step 1” on the softassign algorithm due to the *N-dimensional* procedure must extend a concrete algorithm to compute the joint probability of several graphs. However, both procedures are generic enough to be applied to any other algorithms that use a probabilistic approach, e.g. [8] and [11].

5 Evaluation

We have evaluated the model using a dataset created at the University of Bern [10]. It is composed of 15 capital letters (classes) of the Roman alphabet i.e. A, E, F, H, I, K, L, M, N, T, V, W, X, Y and Z. Each letter is constituted by straight lines which represent edges and terminal points which represent nodes. Nodes are defined over a two-dimensional domain that represents the position (x, y) in the plane. Edges have a one-dimensional and binary attribute that represents the existence or non-existence of a line between two terminal points. Graph-based representations of the prototypes are shown in **Figure 6**. This database contains three sub-databases with different distortion level: low, med, high. Each distortion level of each letter is composed by 150 examples. **Figure 7** shows 3 examples of letter X with low distortion and 3 examples of letter H with high distortion. In this evaluation just high distortion has been used.

To evaluate the new methodologies proposed we have performed two experiments. The aim of the first experiment is to evaluate the consistency of the MI obtained in

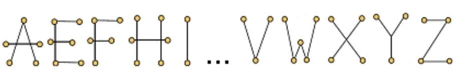


Fig. 6. Graph-based representations of the original prototypes

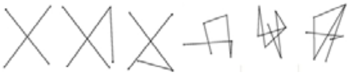


Fig. 7. X and H examples with with low and high distortion level respectively

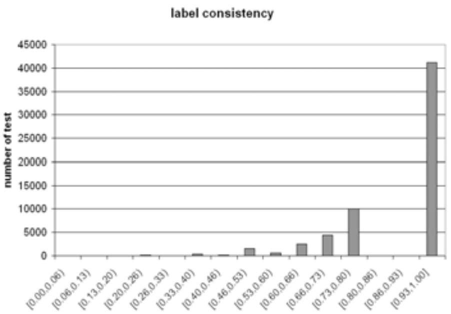


Fig. 8. Label consistency of labelings found using classical softassign

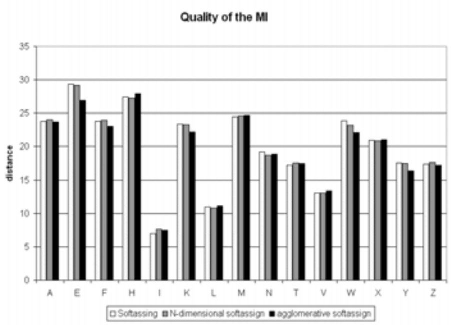


Fig. 9. Quality of the MI using the two algorithms presented and the softassign

step 1 of the basic scheme[1] (Figure 4). The second experiment is addressed to evaluate the goodness of CMI generated by the two schemes (Figure 4 and Figure 5)[2].

To perform the experiment we compute the CMI of a set S composed by three elements. We took 30 AGs of each database class. Using each set, we evaluated all possible labelings resulting 30*29 labelings. With all those labelings, we chose all possible triplets without repeating twice the same graph[3], generating ≈ 4000 triplets. We evaluated the *labeling consistency* for each triplet. The overall results for all test elements are shown in **Figure 8**. It can be observed that approximately 70% of the triplets don't produce labeling errors (the labeling consistency is aprox. 1). The other 30% of the triplets have different levels of consistency.

In the second experiment we compare the mean edit distance [9] of all test elements for each class. We took as a test set the inconsistent triplets found in experiment one. Due to temporal requirement of the *N-dimensional softassign* algorithm, we choose, randomly, a sub-set of 50 elements for each class.

Figure 9 shows the results for the second experiment. We observe that the mean of edit distances is approximately the same for the three methodologies. Therefore, we can conclude that it is possible to eliminate the inconsistencies obtained with the basic scheme without reducing the quality of the MI.

To summarize, it is worth to say that the expected results should reflect that when the basic scheme finds inconsistencies in the labelings, the proposed algorithms reduce to none those inconsistencies at the cost of augmenting the mean distance of the set S. However, we can amazingly see that in most of those cases where the basic scheme finds inconsistencies, the new scheme obtains better MI than the basic scheme while reducing to none the inconsistencies of the MI.

[1] Using the softassign algorithm.

[2] Using the softassign algorithm on the first scheme and the two presented methods in the second scheme.

[3] That is, only result where $(x, y, z) \mid (x \neq y) \wedge (x \neq z) \wedge (y \neq z)$ are produced.

6 Conclusions and Further Work

In this paper, we have presented two new approaches to compute a common labelling between a set of AGs. The main idea of these algorithms is to tackle the problem from the joint-probability point of view. The joint probability represents the probability of all the labelings between all the AGs taken at once. Up to now, the probabilities of each labelling had been considered as independent items and only at the end of the process, when the consistency had to be fulfilled, there where taken all together. Results show that it is possible to remove inconsistencies in marginal labeling function without incrementing the mean distance of the AG set.

Acknowledgements

This research supported by Consolider Ingenio 2010; project CSD2007-00018, by the CICYT project DPI 2007-61452 and by the Universitat Rovira I Virgili (URV) through a PhD research grant.

References

[1] Bonev, B., et al.: Constellations and the unsupervised learning of graphs. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 340–350. Springer, Heidelberg (2007)

[2] Sanfeliu, A., Serratosa, F., Alquézar, R.: Second-Order Random Graphs for modeling sets of Attributed Graphs and their application to object learning and recognition. International Journal of Pattern Recognition and Artificial Intelligence, IJPRAI 18(3), 375–396 (2004)

[3] Serratosa, F., Alquézar, R., Sanfeliu, A.: Function-Described Graphs for modeling objects represented by attributed graphs. Pattern Recognition 36(3), 781–798 (2003)

[4] Wong, A.K.C., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE Transactions on PAMI 7, 599–609 (1985)

[5] Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. Trans. on PAMI 18(4), 377–388 (1996)

[6] Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modelling. In: PH Distributions; ASA, ch. 2, 1st edn., SIAM, Philadelphia (1999)

[7] Solé-Ribalta, A., Serratosa, F.: A structural and semantic probabilistic model for matching and representing a set of graphs, GbR, Venice, Italy. LNCS, vol. 5534, pp. 164–173. Springer, Heidelberg (2009)

[8] Williams, M.L., Wilson, R.C., Hancock, E.R.: Multiple Graph Matching with Bayesian Inference. Pattern Recognition Letters 18, 1275–1281 (1997)

[9] Sanfeliu, A., Fu, K.S.: A Distance Measure Between Attributed Relational Graphs for Pattern Recognition. Trans. Systems, Man, and Cybernetics 13, 353–362 (1983)

[10] Riesen, K., Bunke, H.: Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: SSPR 2008 (2008)

[11] Fekete, G., Eklundh, J.O., Rosenfeld, A.: Relaxation: Evaluation and Applications. Trans. on PAMI 4(3), 459–469 (1981)

[12] Luo, B., Hancock, E.R.: Structural graph matching using the EM algorithm and singular value decomposition. Trans. on PAMI 10(23), 1120–1136 (2001)

Advances in Rotation-Invariant Texture Analysis

Alfonso Estudillo-Romero and Boris Escalante-Ramirez

Universidad Nacional Autonoma de Mexico, Fac. de Ingenieria, Edif. de Posgrado e
Investigacion, Ciudad Universitaria, C.P. 04510, Mexico, D.F., Mexico
aestudillor@uxmcc2.iimas.unam.mx, boris@servidor.unam.mx

Abstract. Robust rotation invariance has been a matter of great interest in many applications which use low-level features such as textures. In this paper, we propose a method to analyze and capture visual patterns from textures regardless their orientation. In order to achieve rotation invariance, visual texture patterns are locally described as one-dimensional patterns by appropriately steering the Cartesian Hermite coefficients. Experiments with two datasets from the Brodatz album were performed to evaluate orientation invariance. High average precision and recall rates were achieved by the proposed method.

Keywords: Texture analysis, steered Hermite transform, image retrieval.

1 Introduction

There is no precise definition of texture, but there are intuitive and interesting properties of texture which are generally assumed by researchers. Texture can be described on a spatial neighborhood, whose size and meaning depend upon the texture type and the scale or resolution at which the texture is perceived. We can also relate texture to a set of repetitive patterns, sometimes named texture primitives, taking place into the neighborhood. Texture gray level values can also form a distribution and characterize the texture. The fact that perception of texture has so many dimensions is an important reason why there is no single technique to represent a variety of textures.

Despite the lack of a precise definition, it is often desired that robust texture analysis achieve one of the most important requirements: to be rotation invariant. Early methods realizing the importance of rotation invariance used polarograms [1] and model based methods proposed a circular symmetric autoregressive model [2] and Gaussian Markov random field models (GMRF)[3].

Recent methods based on Wavelet transforms achieve rotation invariance by performing preprocessing stages over the texture image such as polar transformations [4] or a Radon transform [5]. However, one disadvantage of this strategy is the large number of free parameters and consequently a deeper analysis must be done in order to find optimal parameters for each dataset. On the other hand, some methods achieve rotation invariance by performing post-processing stages such as circular shifts over the feature map according to a dominant orientation [6,7] and addition of the different directional coefficients at each scale of analysis [8].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 145–152, 2009.

In this work we propose a robust method to extract rotation-invariant features from texture images using the steered Hermite transform. The Hermite transform is known to be in agreement with some processes found in early vision systems. Moreover, the steered Hermite transform proposed in [9] and studied in [10] and [11] provides an efficient way to find meaningful patterns at many orientations and then compress them into a few coefficients. Methods using the Discrete Wavelet Transform (DWT) need extra processing stages before or after the feature extraction process in order to achieve rotation invariance, whereas methods based on Gabor wavelets need filters to be tuned at fixed orientations. The property of coefficient steering, based on the directionality of maximum energy, provides a non-fixed orientation filter design which has the advantage of approximately finding the same features regardless the orientation of the input image.

Section 2 summarizes the Hermite transform theory for two-dimensional signals. In Sect. 3 the steered Hermite transform is presented. The proposed methodology is presented in Sect. 4. Experimental results are reported in Sect. 5 and finally, conclusions and future directions are given in Sect. 6.

2 Hermite Transform

For the one dimensional case, a polynomial transform $L_n(x)$ is a local decomposition technique in which an input signal $L(x)$ is localized through a window $V(x)$ and then expanded into orthogonal polynomials $G_n(x)$ at every window position [9]:

$$L_n(x_0) = \int_x L(x) G_n(x_0 - x) V_n^2(x_0 - x) dx \ . \tag{1}$$

The Hermite transform arises when G_n are the Hermite polynomials $H_n(x)$, given by Rodrigues' formula:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n e^{-x^2}}{dx^n}, \quad n = 0, 1, 2, \ldots \ . \tag{2}$$

and the orthogonal window corresponds to a Gaussian window:

$$V(x) = \frac{1}{\sqrt{\sqrt{\pi}\sigma}} \cdot e^{-x^2/2\sigma^2} \ . \tag{3}$$

Following (1), the expansion coefficients $L_n(x)$ can be derived by convolution of the signal $L(x)$ with the Hermite analysis functions $d_n(x)$, which are given in terms of the window and Hermite polynomials as:

$$d_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \cdot \frac{1}{\sigma\sqrt{\pi}} H_n\left(\frac{x}{\sigma}\right) e^{-x^2/\sigma^2} \ . \tag{4}$$

The Hermite analysis functions can be easily generalized to two dimensions because of the property of being both spatially separable and rotationally symmetric. We then can write the two dimensional analysis functions as:

$$d_{n-m,m}(x, y) = d_{n-m}(x) d_m(y) \ . \tag{5}$$

where $n-m$ and m denote the analysis order in x and y direction respectively. As a result, we can expand a given input image $L(x, y)$ into the basis $d_{n-m,m}(x, y)$ as:

$$L_{n-m,m}(x_0, y_0) = \int_x \int_y L(x, y) d_{n-m,m}(x_0 - x, y_0 - y) dx dy \ . \tag{6}$$

for $n = 0, 1, \ldots, \infty$ and $m = 0, \ldots, n$.

3 Steered Hermite Transform

A steerable filter is described as a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of *basis filters* [12]. Since all Hermite filters are polynomials times a radially symmetric window function, rotated versions of a filter of order n can be constructed by taking linear combinations of the original filters of order n. In this way, a more general expression of the original $L_{n-m,m}$ Cartesian Hermite coefficients can be written in terms of the orientation selectivity θ [11]:

$$L^{\theta}_{n-m,m}(x_0, y_0, \theta) = \sum_{k=0}^{n} L_{n-k,k}(x_0, y_0) \alpha_{n-k,k}(\theta) \ . \tag{7}$$

which has been named the steered Hermite transform in [10]. The terms $\alpha_{n-m,m}(\theta)$ are the Cartesian angular functions of order n which give such orientation selectivity are defined as:

$$\alpha_{n-m,m}(\theta) = \sqrt{C_n^m} \cos^{n-m}(\theta) \sin^m(\theta) \ . \tag{8}$$

Considering the ideally rotation of an input image within a circularly symmetric window and assuming that no artifacts are introduced due to interpolation and discretization, we then can assume that there is no lost of information during the rotation process. If this is the case, energy is preserved for each rotated image. Thus, we can write the local energy in terms of the steered Hermite coefficients as:

$$E_N = \sum_{n=0}^{N} \sum_{m=0}^{n} [L_{n-m,m}]^2 = \sum_{n=0}^{N} \sum_{m=0}^{n} [L^{\theta}_{n-m,m}]^2 \ . \tag{9}$$

for all $N \geq 0$. In natural images, many of the image details that are of prime importance, such as edges and lines, can be locally described as one-dimensional patterns, that is, patterns that vary only in one direction (and are constant along the orthogonal direction). One may distinguish 1D local energy terms and 2D local energy terms. Thus, we can split local energy of (9) up to order N as:

$$E_N = [L_{0,0}]^2 + E_N^{1D} + E_N^{2D} \ . \tag{10}$$

where $L_{0,0}$ represents the DC Hermite coefficient and

$$E_N^{1D} = \sum_{n=1}^{N} [L^{\theta}_{n,0}]^2 \ . \tag{11}$$

$$E_N^{2D} = \sum_{n=1}^{N} \sum_{m=1}^{n} [L_{n-m,m}^{\theta}]^2 \ . \tag{12}$$

One of the objectives when steering coefficients is to maximize detection of patterns along a given local direction θ. In this way, [10], [11] and [13] propose strategies in which θ is selected such that E_N^{1D} is maximized. As a consequence, compaction of energy (i.e Hermite coefficients) can be efficiently achieved.

4 Proposed Methodology

Figure 1 summarizes the proposed methodology to extract rotation-invariant texture patterns based on the stereed Cartesian Hermite coefficients. Note that this is a general scheme, no classification methods neither distance metrics are involved, leaving this interesting areas open to future investigations. Moreover different texture features are suitable to be extracted from the steered Hermite coefficients.

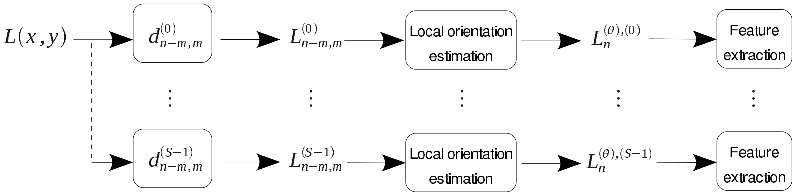


Fig. 1. Texture feature extraction methodology using steered Hermite transform

In this work we used the mean and standard deviation features which seem to better represent on a global sense the behavior of texture patterns distributions for a given frequency (order n) and scale of analysis. The feature vector is formed by concatenating mean and standard deviation for each steered Hermite coefficient $1 \leq n \leq N$ at every scale of analysis s, where $0 \leq s \leq S-1$ and S represents the number of scales:

$$\boldsymbol{f} = [\mu_1^{(0)}, \sigma_1^{(0)}, \ldots, \mu_N^{(0)}, \sigma_N^{(0)}, \ldots, \mu_1^{(1)}, \sigma_1^{(1)}, \ldots, \mu_N^{(S-1)}, \sigma_N^{(S-1)}] \ . \tag{13}$$

5 Experimental Results

The purpose of the following experiments was to evaluate the ability to extract rotation-invariant visual patterns from texture images with the proposed methodology. Two experiments proposed in early works were reproduced. Although experiments can hold for a particular application such as image retrieval or indexing, our principal contribution is a method to analyze and capture visual patterns regardless their orientation.

Evaluation of texture analysis methods is frequently presented as the behavior of the average of both precision and recall as functions of a requested (query) number of images. Let n_g be the number of "ground truth" texture images for the class c and let n_k be the correct number of retrieved "ground truth" texture images when k queries are performed. Then, precision is defined as:

$$P_c = \frac{n_k}{k} \quad . \tag{14}$$

and recall as:

$$R_c = \frac{n_k}{n_g} \quad . \tag{15}$$

Note that perfect scores precision and recall are obtained when the number of delivered "ground truth" texture images equals the number of queries. By computing the average of both precision and recall for each class and for different k queries it is possible to evaluate the robustness of the rotation-invariant texture analysis method.

The Brodatz texture image dataset [14] was used in both experiments. A Hermite decomposition was performed up to $N = 8$ with four scales of analysis. Thus, a feature vector of 64 elements was formed by concatenating mean and standard deviation for each steered Hermite coefficient.

The similarity measure was obtained by computing the distance between the feature vectors using the Canberra distance metric:

$$d = \sum_{i=1}^{z} \frac{|\boldsymbol{f}_i - \boldsymbol{g}_i|}{|\boldsymbol{f}_i| + |\boldsymbol{g}_i|} \quad . \tag{16}$$

Experiment I. For this experiment, following the configuration of dataset 4 presented in [4], an image dataset of 25 texture classes from the Brodatz album was prepared. First, each 512×512 texture image was rotated with 36 equally spaced angles, ranging from 0 to $35\pi/36$ radians with incremental step size of $\pi/36$. We do not rotate beyond the upper limit $35\pi/36$ radians because it would have redundant rotated texture images. Each rotated texture image is then partitioned from the center of the image to reach 128×128 pixels. As a result 36 rotated texture images comprise the "ground truth" images for each texture class and a dataset of 25×36 texture images was formed with rotation of a single area.

Figure 2 shows graphs of the precision and recall average rates. We noted how well the steered Hermite transform performed for the majority of 36 "ground truth" texture images, that is, 99.88% retrieval performance when $P(\%) = R(\%)$ (i.e. for $k = 36$ first retrieved texture images). Comparing our results with the presented in [4], for dataset with rotation only from a single area, we obtained similar results (P=100%) even for all the first $k = 32$ retrieved texture images.

Experiment II. For this experiment dataset of experiment I was extended to 108 Brodatz texture images. First, each 512×512 texture image was rotated with 16 equally spaced angles, ranging from 0 to $15\pi/16$ radians with incremental step size of $\pi/16$ as proposed in [8]. Each rotated texture image is then partitioned

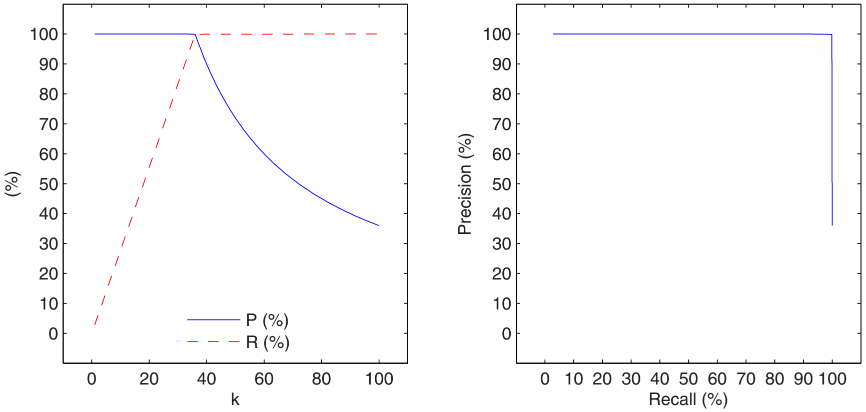


Fig. 2. Average retrieval performance. Note that approximately a perfect average compromise between precision and recall (99.88%) was obtained when retrieving the first 36 "ground truth" texture images.

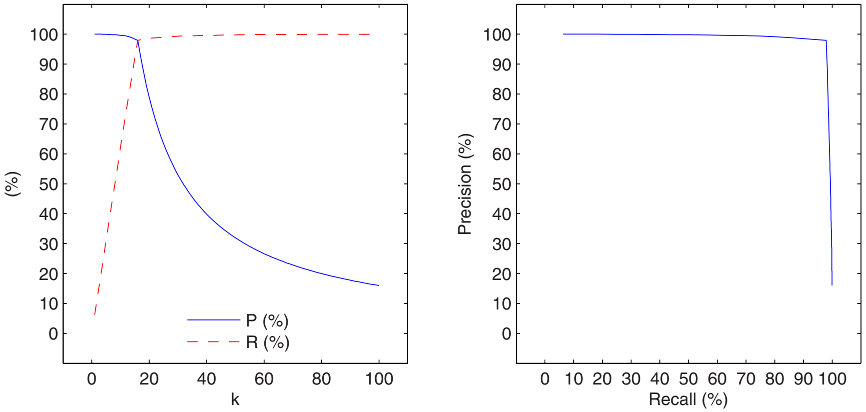


Fig. 3. Average retrieval performance of experiment II. Note that approximately a perfect average compromise between precision and recall (97.92%) was obtained when retrieving the first 16 "ground truth" texture images.

from the center of the image to reach 128×128 pixels. As a result 16 rotated texture images comprise the "ground truth" images for each texture class and a dataset of 108×16 texture images was formed with rotation of a single area.

Figure 3 shows graphs of the precision and recall average rates. The steered Hermite transform-based features got performance retrieval of 97.93% when $P(\%) = R(\%)$ (i.e. for $k = 16$ first retrieved texture images). In [8] and [6] experiments were performed with similar texture images datasets. Han and Ma [8] propose a Gabor-based rotation-invariant method and compare it with previous Gabor-based methods. In [6] a rotation-invariant method using wavelet-based hidden Markov trees was proposed and reported an average retrieval rate of 91.25%. Comparing results, it seems that the proposed methodology outperformed average retrieval rates of the methods above mentioned. We noticed in our experiment that for all the first $k = 3$ retrieved texture images the average precision rate was P=100%.

6 Conclusions

In this work, a rotation-invariant texture analysis methodology was proposed. Texture analysis was performed taking into consideration visual information from the texture images. First, the analysis functions of the Hermite transform performed filtering and extracted visual details which were then locally described as one-dimensional patterns by appropriately steering the Cartesian Hermite coefficients. Mean and standard deviation were computed from each steered Hermite coefficient and concatenated to form a vector of 64 features.

Results showed that important visual patterns are well extracted after the steering of Cartesian Hermite coefficients regardless their orientation. Moreover, we observed an important comprise between average precision and recall rates with the present method outperforming results previously reported with other methods.

Although evaluation of the proposed method was conducted for texture image retrieval, our principal contribution is a method to analyze and capture visual patterns regardless their orientation. Therefore, many applications that use texture features can be implemented following the proposed scheme. Future works will include classification and segmentation evaluations.

Acknowledgements

This work was supported by UNAM-PAPIIT IN106608.

References

1. Davis, L.S.: Polarograms: A new tool for image texture analysis. Pattern Recognition 13(3), 219–223 (1981)
2. Kashyap, R.L., Khotanzad, A.: A model-based method for rotation invariant texture classification. IEEE Trans. PAMI 8(4), 472–481 (1986)
3. Cohen, F., Fan, Z., Patel, M.: Classification of rotated and scaled textured images using gaussian markov random field models. IEEE Trans. PAMI 13(2), 192–202 (1991)
4. Pun, C.M., Lee, M.C.: Log-polar wavelet energy signatures for rotation and scale invariant texture classification. IEEE Trans. PAMI 25(5), 590–603 (2003)
5. Jafari-Khouzani, K., Soltanian-Zadeh, H.: Rotation-invariant multiresolution texture analysis using radon and wavelet transforms. IEEE Transactions on Image Processing 14(6), 783–795 (2005)
6. Rallabandi, V.R., Rallabandi, V.S.: Rotation-invariant texture retrieval using wavelet-based hidden markov trees. Signal Processing 88(10), 2593–2598 (2008)

7. Montoya-Zegarra, J.A., Papa, J.P., Leite, N.J., da Silva Torres, R., Falco, A.X.: Learning how to extract rotation-invariant and scale-invariant features from texture images. EURASIP Journal on Advances in Signal Processing 691924 (2008)
8. Han, J., Ma, K.K.: Rotation-invariant and scale-invariant gabor features for texture image retrieval. Image and Vision Computing 25(9), 1474–1481 (2007)
9. Martens, J.B.: The hermite transform-theory. IEEE Transactions on Acoustics, Speech and Signal Processing 38(9), 1595–1606 (1990)
10. van Dijk, A.M., Martens, J.B.: Image representation and compression with steered hermite transforms. Signal Processing 56(1), 1–16 (1997)
11. Silvan-Cardenas, J., Escalante-Ramirez, B.: The multiscale hermite transform for local orientation analysis. IEEE Transactions on Image Processing 15(5), 1236–1253 (2006)
12. Freeman, W., Adelson, E.: The design and use of steerable filters. IEEE Trans. PAMI 13(9), 891–906 (1991)
13. Martens, J.B.: The Hermite transform: a survey. EURASIP Journal of Applied Signal Processing 26145 (2006)
14. Brodatz, P.: Texture: a photographic album for artists and designers. Dover, New York (1966)

SAR Image Segmentation Using Level Sets and Region Competition under the $\mathcal{G}^H$ Model

Maria Elena Buemi, Norberto Goussies, Julio Jacobo, and Marta Mejail

Departamento de Computación, Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires,
Ciudad Universitaria, Pabellón I, 1428 Ciudad de Buenos Aires, República Argentina
{mebuemi,ngoussie,jacobo,marta}@dc.uba.ar
http://www-2.dc.uba.ar/grupinv/imagenes/

Abstract. Synthetic Aperture Radar (SAR) images are dificult to segment due to their characteristic noise, called *speckle*, which is multiplicative, non-gaussian and has a low signal to noise ratio. In this work we use the $\mathcal{G}^H$ distribution to model the SAR data from the different regions of the image. We estimate their statistical parameters and use them in a segmentation algorithm based on multiregion competition. We then apply this algorithm to segment simulated as well as real SAR images and evaluate the accuracy of the segmentation results obtained.

Keywords: SAR images, $\mathcal{G}^H$distribution, multiregion competition, level set, segmentation.

1 Introduction

Several types of imaging devices employ coherent illumination as, for instance, Synthetic Aperture Radar (SAR), sonar, laser and ultrasound-B. The images generated by these devices are affected by a noise called speckle, a kind of degradation that does not obey the classical hypotheses of being Gaussian and additive. Speckle noise reduces the ability to extract information from the data, so specialized techniques are required to deal with such imagery. Identifying boundaries that separate different areas is one of the most important image understanding goals. High level image processing relies on precise and accurate boundaries, among other features. Finding boundaries between regions of different roughness is a hard task when data are contaminated by speckle noise. Speckled data can be statistically modeled using the family of $\mathcal{G}$ distributions [1], since these probability laws are able to describe the observed data better than other laws, specially in the case of rough and extremely rough areas. As a case of interest, in SAR images such situations are common when scanning urban spots or forests on undulated relief, and for them the more classical Γ and $\mathcal{K}$ distributions do no exhibit good performance [1,2]. Under the $\mathcal{G}$ model, regions with different degrees of roughness can be characterized by the statistical parameters. Therefore, this information can be used to find boundaries between regions with different textures. The propose of this work is to use region competition methods

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 153–160, 2009.

under $\mathcal{G}^H$. We replace the hypothesis of Gaussian noise in the region competition functional [3] with the hypothesis of $\mathcal{G}^H$ distributed noise. The minimization of the resulting functional is performed via the level set formalism [4]. This work is structured as follows, in section 2 we describe the $\mathcal{G}^H$ distribution, in section 3 we describe the segmentation process based on region competition and level set minimization, in section 4 we present the results on simulated and real images and our conclusions.

2 Image Model and the $\mathcal{G}^H$ Distribution

Monopolarized SAR images can be modeled as the product of two independent random variables: one corresponding to the backscatter X, which is a physical quantity that depends on the geometry and the electromagnetic characteristics of the sensed surface, and the other one corresponding to the speckle noise Y, the typical noise of coherent illumination devices. In this manner

$$Z = X \cdot Y \tag{1}$$

models the return Z in each pixel under the multiplicative model. For monopolarized data, the speckle noise Y is modeled as a $\Gamma(n,n)$ distributed random variable, where n is the number of looks, so its density is given by

$$f_Y(y) = \frac{n^n}{2^n \Gamma(n)} y^{n-1} \exp\left(-\frac{1}{2} n y\right),\ y > 0. \tag{2}$$

Also for this type of data, the backscatter X is considered to obey a Generalized Inverse Gaussian law, denoted as $\mathcal{N}^{-1}(\alpha, \lambda, \gamma)$ [5]. This distribution has been proposed as a general model for backscattering, its density function being

$$f_X(x) = \frac{(\lambda/\gamma)^{\alpha/2}}{2K_\alpha\left(\sqrt{\lambda\gamma}\right)} x^{\alpha-1} \exp\left(-\frac{1}{2}\left(\lambda x + \frac{\gamma}{x}\right)\right),\ x > 0. \tag{3}$$

The values of the statistical parameters γ, λ and α are constrained to be: $\gamma > 0$ and $\lambda \geq 0$ when $\alpha < 0$, $\gamma > 0$ and $\lambda > 0$ when $\alpha = 0$, and $\gamma \geq 0$ and $\lambda > 0$ when $\alpha > 0$. The function K_α is the modified Bessel function of the third kind.

The backscatter X can exhibit different degrees of roughness and therefore, considering this characteristic, it could follow different models.

For smooth areas, such as pastures and many types of crops, a constant $\mathcal{C}$ distribution is an adequate model for X. For homogeneous and also for moderately heterogeneous areas, the Γ distribution is a good model, and the corresponding distribution for the SAR data Z is the $\mathcal{K}$ distribution .

In order to model a wide range of targets, ranging from rough to extremely rough targets, the reciprocal of Gamma Γ^{-1} [1] and the Inverse Gaussian $IG(\gamma, \lambda)$ distributions can be used. This in turn results in the $\mathcal{G}^0$ [6,7,8,2,9], and the $\mathcal{G}^H$ distributions for the return Z, respectively. These distributions have the additional advantage of their mathematical tractability, when compared to the $\mathcal{K}$ distribution.

In this paper, we propose the use of the Inverse of Gamma distribution to model the backscatter X. This statistical law is the result of making $\alpha = -1/2$ in the Generalized Inverse Gaussian distribution $\mathcal{N}^{-1}(\alpha, \lambda, \gamma)$ so it becomes the $IG(\gamma, \lambda)$ distribution.

The density function of this distribution is given by (Eq. 4).

$$f_X(x) = \sqrt{\frac{\gamma}{2\pi x^3}} \exp\left(-\frac{\left(\sqrt{\lambda}x - \sqrt{\gamma}\right)^2}{2x}\right),\ x > 0, \tag{4}$$

with $\lambda, \gamma > 0$. The parameters γ and λ can be used to define a new pair of parameters ω and η, given by $\omega = \sqrt{\gamma\lambda}, \eta = \sqrt{\gamma/\lambda}$ so formula (4) can be rewritten as

$$f_X(x) = \sqrt{\frac{\omega\eta}{2\pi x^3}} \exp\left(-\frac{1}{2}\omega\frac{(x-\eta)^2}{x\eta}\right),\ x > 0. \tag{5}$$

So $X \sim IG\left(\omega, \eta\right)$, and it is possible to see that the corresponding moments are

$$\mathbb{E}\left[X^r\right] = \sqrt{\frac{2\omega}{\pi}} \exp\left(\omega\right)\eta^r K_{r-\frac{1}{2}}\left(\omega\right). \tag{6}$$

where $K_{r-\frac{1}{2}}\left(\omega\right)$ is the modified Bessel function of the third kind. Given that the order of this function is $r - \frac{1}{2}$ with r an integer number, there is a closed formula that allows it to be easily evaluated.

The corresponding density function for the return Z, is given by

$$\begin{aligned} f_{\mathcal{G}^H}(z) = \tfrac{n^n}{\Gamma(n)}\sqrt{\tfrac{2\omega\eta}{\pi}}\exp(\omega)\left(\tfrac{\omega}{\eta(\omega\eta+2nz)}\right)^{n/2+1/4} \\ .\ z^{n-1}K_{n+1/2}\left(\sqrt{\tfrac{\omega}{\eta}(\omega\eta+2nz)}\right), \end{aligned} \tag{7}$$

with $\omega, \eta, z > 0$ and $n \geq 1$, respectively.

The moments of the $\mathcal{G}^H$ distribution are

$$E_{\mathcal{G}^H}(Z^r) = \left(\frac{\eta}{n}\right)^r \exp\left(\omega\right)\sqrt{\frac{2\omega}{\pi}}K_{r-1/2}(\omega)\frac{\Gamma(n+r)}{\Gamma(n)}, \tag{8}$$

and are used to estimate the statistical parameters.

3 Image Segmentation

Let $I : \Omega \rightarrow \Re$ be an image defined over $\Omega \subseteq \Re^2$. The goal of the segmentation process is to find a family of regions $\mathcal{R} = \{\mathbf{R}_i\}_{i=1\ldots N}$ such that:

- Each region is a subset of the image domain $\mathbf{R}_i \subseteq \Omega$.
- The regions are pairwise disjoint $\mathbf{R}_i \cap \mathbf{R}_j = \phi\ \forall i \neq j$.
- Cover the image domain $\cup_{i=1}^{N}\mathbf{R}_i \subseteq \Omega$.
- The points in each region share some image characteristics.

In [3] Zhu and Yuille proposed that the intensity values of the points inside each region are consistent with having been generated by one of a family of prespecified probability distributions $P(I(\boldsymbol{x}) : \boldsymbol{\alpha_i})$, where $\boldsymbol{\alpha_i}$ are the parameters of the distribution for the region $\mathbf{R}_i$. In [3] the image segmentation problem is posed as the minimization of the energy functional:

$$E^{ZY}(\mathbf{R}_1, ..., \mathbf{R}_N, \boldsymbol{\alpha_1}, ..., \boldsymbol{\alpha_N}) = \sum_{i=1}^{N} \left(- \int_{\mathbf{R}_i} \log P(I(\boldsymbol{x}) : \boldsymbol{\alpha_i}) d\boldsymbol{x} + \frac{\mu}{2} \oint_{\partial \mathbf{R}_i} ds \right) \tag{9}$$

being $\partial \mathbf{R}_i$ the boundary of the region $\mathbf{R}_i$. The regions that minimizes the functional are the desired family of regions $\mathcal{R}$. The first term, is the sum of the cost for coding the intensity of every $\boldsymbol{x}$ pixel inside the $\mathbf{R}_i$ according to it's distribution. The second term, is a regularization term and penalizes large boundaries. The parameter $\mu > 0$ is a weighting constant controlling the regularization. In this work we assume that $I(\boldsymbol{x}) \sim \mathcal{G}^H$, therefore $P(I(\boldsymbol{x}) : \boldsymbol{\alpha_i})$ is given by Eq. (7) and $\boldsymbol{\alpha_i} = (\omega_i, \eta_i)$.

3.1 Level Sets Based Minimization

Although the suggested functional in Eq. (9) describes the problem quite accurately, their minimization is very difficult. Level sets based methods are a way to solve this problem. The methods has a lot of attractive properties. First, level sets can describe topological changes in the segmentation. Second, it is not necessary to discretisize the contours of the objects.

Level sets [4] based methods to minimize functionals like Eq. (9) has been addressed by multiple works [10,11,12,13]. Most of them uses more than one level set function to represent the regions. The main difficulty is that the evolution of level set functions need to be coupled in order respect the restrictions of disjoint regions. In the two-region case this constraint is implicitly satisfied.

In [10] Chan and Vese extended the work in [14] to deal multiple regions using only $\log N$ level set functions. When the number of regions is a power of 2, this model implicitly respect the restriction that the regions are disjoint. However when the number of level set functions is not a power of two this model shows some problems. The first problem is that region boundaries are weighted twice. The second problem is that the model introduces empty regions.

A different approach is proposed in [13] where $N - 1$ level set functions $\{\Phi_i\}_{i=1...N-1}$ are used to represent N regions. In the work they define the regions $R_{\Phi_i} = \{x \in \Omega | \Phi(x) > 0\}$ and the desired segmentation is given by the family $\mathcal{R} = \left\{ R_{\Phi_1}, R^c_{\Phi_1} \cap R_{\Phi_2}, R^c_{\Phi_1} \cap R^c_{\Phi_2} \cap R_{\Phi_3}, ..., R^c_{\Phi_1} \cap ... \cap R^c_{\Phi_{N-1}} \right\}$ which satisfies the partition constraint by definition. The proposed coupled motion equations are:

$$\frac{\partial \Phi_j}{\partial t}(\boldsymbol{x}, t) = ||\nabla \Phi_j(\boldsymbol{x}, t)|| \left(P(I(\boldsymbol{x}) : \alpha_j) - \psi_j(\boldsymbol{x}) + \mu \mathsf{div} \left(\frac{\nabla \Phi_j(\boldsymbol{x}, t)}{||\nabla \Phi_j(\boldsymbol{x}, t)||} \right) \right) \tag{10}$$

with $1 \leq j \leq N - 1$, and where $\psi_j(\boldsymbol{x})$ is given by:

$$
\begin{aligned}
\psi_j(\boldsymbol{x}) = {} & P(I(\boldsymbol{x}) : \alpha_{j+1})\chi_{R_{\Phi_{j+1}}}(\boldsymbol{x}) \\
& + P(I(\boldsymbol{x}) : \alpha_{j+2})\chi_{R^c_{\Phi_{j+1}} \cap R_{\Phi_{j+2}}}(\boldsymbol{x}) \\
& \dots \\
& + P(I(\boldsymbol{x}) : \alpha_{N-1})\chi_{R^c_{\Phi_{j+1}} \cap \dots \cap R^c_{\Phi_{N-2}} \cap R_{\Phi_{N-1}}}(\boldsymbol{x}) \\
& + P(I(\boldsymbol{x}) : \alpha_{N})\chi_{R_{\Phi^c_{j+1}} \cap \dots \cap R^c_{\Phi_{N-2}} \cap R_{\Phi^c_{N-1}}}(\boldsymbol{x})
\end{aligned}
$$

The last approach described is simple and easy to implement. It has been successfully used in SAR segmentation images in the work [15]. Thereof this is the approach we have adopted in our work.

4 Results and Conclusions of Image Segmentation Using G^H Models

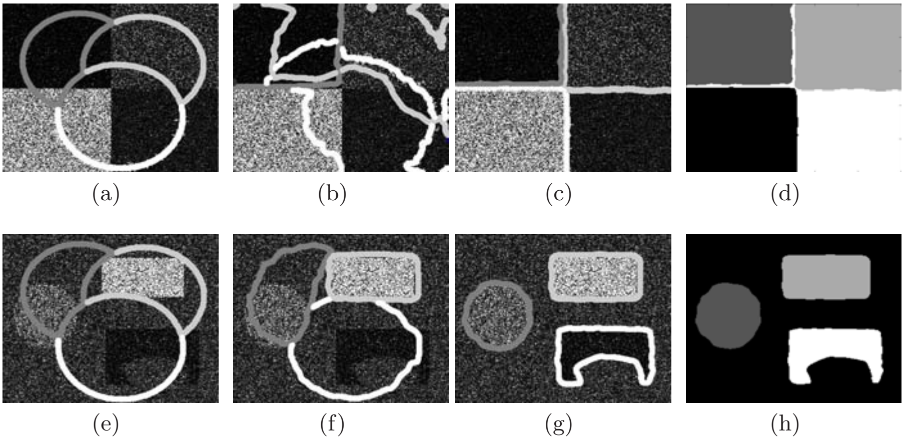
The proposed algorithm has been tested on a range of simulated and real images. The results for two simulated images are shown in Fig. 1 (a) through (d) and Fig. 1 (e) through (h), showing from left to right, the initial contours, their evolution and final results. These images were generated using the $\mathcal{G}^H$ distribution. The parameters used to generate each of the regions and their corresponding estimates from the segmented images are shown in Table 1. The percentage of pixels correctly classified in the first image is 97.05% and in the second image is 96.89%. The obtained results for the segmentation of simulated images are similar in performance to those obtained by [15].

Table 1. Values for the parameters used to generate the simulated data in Fig. 1(a) and Fig. 1(e) and their corresponding estimates, calculated from the segmented regions depicted in Fig. 1(c) and Fig. 1(g)

	Figure 1(a)				Figure 1(e)			
Region color	η	ω	η-estimate	ω-estimate	η	ω	η-estimate	ω-estimate
background	2.75	57.60	2.47	55.8	13.4	7.4	13.62	7.37
dark gray	3.1	10.5	2.88	10.57	1.95	67.60	1.33	56.97
light gray	1.08	2.25	1.13	2.20	8.1	15.50	7.71	15.14
white	10.0	5.0	7.07	4.90	1.43	3.16	1.51	3.15

Table 2. Estimated $\mathcal{G}^H$ parameters for the segmented regions shown in Fig. 2(c) and Fig. 2(g)

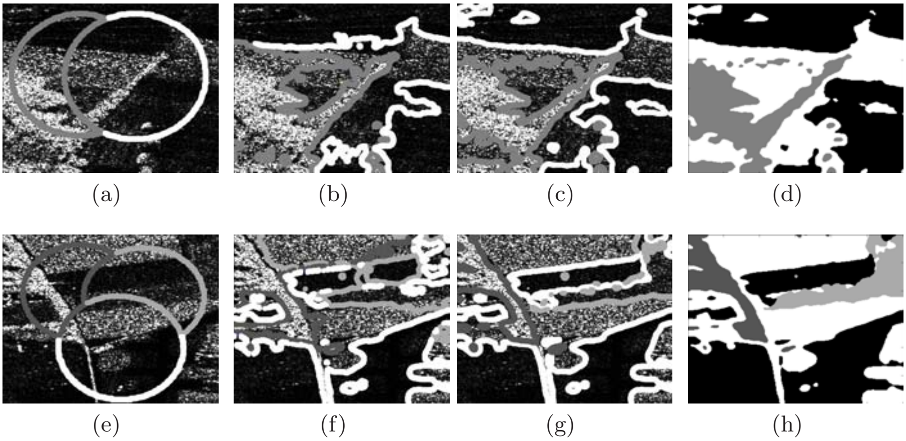
	Figure 2(a)		Figure 2(e)	
Region color	η	ω	η	ω
background	2.60	3.38	1.06	2.09
dark gray	12.93	2.95	3.21	17.46
light gray	66.85	3.08	46.54	0.62
white	—-	—-	9.14	2.35



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 1. Results for two simulated images. (From left to right) Column 1: initial curves, Column 2: position of curves at iteration 11, Column 3: final position of curves, Column 4: segmentation result. The segmented regions and their corresponding contours are shown with the same gray level.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. Results for two real images: (From left to right), Column 1: initial curves, Column 2: position of curves at iteration 11, Column 3: final position of curves, Column 4: segmentation result. The segmented regions and their corresponding contours are shown with the same gray level.

Results for two real images with different number of regions are shown in Fig. 2 (a) through (d) and Fig. 2 (e) through (h), here again showing, from left to right, the initial contours, their evolution and final results. These real images were extracted from an E-SAR image acquired over the DLR location at Oberpfaffenhofen, Germany. The estimated parameters are shown in Table 2. The number of regions used in the segmentation of each of the images were estimated by visual inspection. The results obtained on these images show good segmentation performance for the proposed method. As an example of this, we can exhibit the dark gray region in the segmentation result of the Fig. 2(e) which has a small estimated value for the statistical parameter ω, meaning that there should be buildings in that region. This can be confirmed from a visual inspection of maps of this area.

The presented results support the idea that characterization of regions in SAR images through the use of statistical parameters of the $\mathcal{G}^H$ distribution is very useful and it can be incorporated succesfully in a level set based segmentation scheme.

References

1. Frery, A.C., Müller, H.-J., Yanasse, C.C.F., Sant'Anna, S.J.S.: A model for extremely heterogeneous clutter. IEEE Transactions on Geoscience and Remote Sensing 35(3), 648–659 (1997)
2. Mejail, M.E., Frery, A.C., Jacobo-Berlles, J., Bustos, O.H.: Approximation of distributions for SAR images: proposal, evaluation and practical consequences. Latin American Applied Research 31, 83–92 (2001)
3. Zhu, S.C., Yuille, A.: Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 884–900 (1996)
4. Sethian, J.A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, Cambridge (2007)
5. Frery, A.C., Correia, A., Renno, C.D., Freitas, C.D.C., Jacobo-Berlles, J., Vasconcellos, K.L.P., Mejail, M., Sant'Anna, S.J.S.: Models for synthetic aperture radar image analysis. Resenhas (IME-USP) 4, 45–77 (1999)
6. Jacobo-Berlles, J., Mejail, M., Frery, A.C.: The ga0 distribution as the true model for sar images. In: SIBGRAPI 1999: Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing, Washington, DC, USA, pp. 327–336. IEEE Computer Society, Los Alamitos (1999)
7. Mejail, M., Frery, A.C., Jacobo-Berlles, J., Kornblit, F.: Approximation of the ka distribution by the ga. In: Second Latinoamerican Seminar on Radar Remote Sensing: Image Processing Techniques, pp. 29–35 (1999)
8. Mejail, M.E., Jacobo-Berlles, J., Frery, A.C., Bustos, O.H.: Classification of sar images using a general and tractable multiplicative model. International Journal of Remote Sensing 24(18), 3565–3582 (2003)
9. Quartulli, M., Datcu, M.: Stochastic geometrical modelling for built-up area understanding from a single SAR intensity image with meter resolution. IEEE Transactions on Geoscience and Remote Sensing 42(9), 1996–2003 (2004)
10. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision 50, 271–293 (2002)
11. Chung, G., Vese, L.: Energy minimization based segmentation and denoising using a multilayer level set approach. Energy Minimization Methods in Computer Vision and Pattern Recognition, 439–455 (2005)

12. Brox, T., Weickert, J.: Level set segmentation with multiple regions level set segmentation with multiple regions. IEEE Transactions on Image Processing 15(10), 3213–3218 (2006)
13. Mansouri, A.R., Mitiche, A., Vazquez, C.: Multiregion competition: A level set extension of region competition to multiple region image partitioning. Computer Vision and Image Understanding 101(3), 137–150 (2006)
14. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Transactions on Image Processing 10(2), 266–277 (2001)
15. Ayed, I.B., Mitiche, A., Belhadj, Z.: Multiregion level-set partitioning of synthetic aperture radar images. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(5), 793–800 (2005)

A New Segmentation Approach for Old Fractured Pieces

Jesus Llanes[1], Antonio Adan[1], and Santiago Salamanca[2]

[1] Escuela Superior de Informática. Universidad de Castilla-La Mancha,
13071 Ciudad Real, Spain
{Jesus.Llanes,Antonio.Adan}@uclm.es
[2] Escuela de Ingenierías Industriales. Universidad de Extremadura,
06006 Badajoz, Spain
ssalaman@unex.es

Abstract. In this paper we propose a method for characterizing the surface of fractured pieces which come from old and complete original objects. Natural fractured pieces are difficult to segment due to the fact that the faces and edges are not well defined. For this reason, standard local feature based approaches are clearly inefficient to make an efficient segmentation. Our segmentation procedure is based on the Cone Curvature (CC) concept applied over the original dense models which provide standard scanner modeling tools. This CC based-method allows us to explore the surface from multiple neighborhood levels and to find a compact segmentation solution which characterizes different parts of the piece. A wide experimentation has been carried out on a set of old fractured pieces belonging to the remains of roman sculptures.

Keywords: 3D segmentation, 3D shape analysis, 3D data processing.

1 Introduction

The automatic reconstruction of fragmented objects through the matching of its fragments is a very common problem in archaeology, paleontology and art restoration. This is a challenging problem which has not been completely solved yet. It is possible to manually reconstruct; however, it may be a very tedious and difficult task for humans. For this reason, it is crucial to find methods that allow executing these tasks by help of a computer. The computer should select correct combinations of fragments on its own and yield coherent reconstructions.

Medium and high level 3D data processing (segmentation, labeling, recognition and understanding) over old fractured pieces is a very difficult field that remains currently under research. This is due to the fact that the original geometry has changed over ages without following a specific manipulation pattern turning into non-usual geometry. Therefore, most of conventional techniques applied on standard models are inefficient in this environment. Figure 1 shows a set of old pieces which we are currently working with.

The segmentation of the fragments into different faces and boundaries is one of the key steps when a reconstruction is carried out. In this area, many authors have proposed segmentation algorithms which work with usual objects ([1], [2], [3]). Most techniques are based on edge detection, region growing and probabilistic methods.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 161–168, 2009.

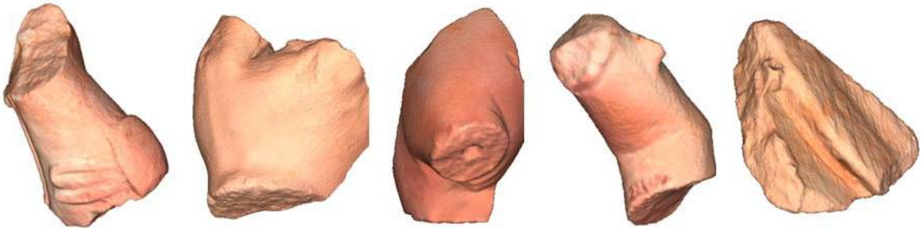


Fig. 1. Tridimensional models corresponding to a set of old fragments belonging to the Roman Art Museum of Merida (Spain)

Edge detection techniques are those which allow detecting the edges of an object to segment it in faces. Several representative works follow.

Liu [4] proposed a method for 3D surface segmentation algorithm based on the classical Robert's operator into 3D space, which works by computing the sum of the squares of the differences between diagonally adjacent pixels. Zucker and Hummel [5], developed an algorithm to perform an optimal three dimensional edge detection based on a Sobel operator which extract the edges using the magnitude of the image gradient at each node of the object.

On the other hand Huang et al [6] perform the segmentation of the object into faces using a multi – scale edge extraction. The algorithm that implements the multi – scale edge extraction is based on the curvature of the surface, measured by the integral invariants [7]. Bellon and Silva [8] perform an object segmentation method based on the edge extraction. The algorithm detects de edge points by the comparison between the normal angles of the surface points of the object. Gotardo et al. [9] proposes a segmentation method which classifies the surfaces points as flat or curved. To make this the algorithm calculates the angular change in the normal direction in moving from a point to nearby points.

Region growing techniques are based on extracting a region of an object using some predefined criteria. From this seed point, the algorithm grows finding connecting areas that fit to the predefined criteria.

In Papaioannou et al [10], [11], [12], segmentation is performed by a region growing method based on the average normal of the polygons that form the mesh. The procedure begins with an arbitrary polygon. The adjacent polygons are classified as related to the same face if their average normal do not deviate more than a predefined threshold.

Generally, the method used to segment archaeological objects is based on edges detection. However, it is very difficult to detect the edges due to the fact that some of the fragments have very smooth borders because of erosion. In Figure 2a) some fragments belonging to a set of archaeological finds are shown. The smooth segments of the borders are marked. When an edge detection algorithm is applied to an object that has smooth borders, it is not impossible to detect them correctly. We applied an algorithm to detect the edges based on the simple curvature values of the mesh nodes. The edge nodes detected by this algorithm are shown in Figure 2 c). It could be noted that the algorithm did not detect any point in the segment of the border that is especially smooth. If we try to define a region delimited by the extracted edges, the system fails

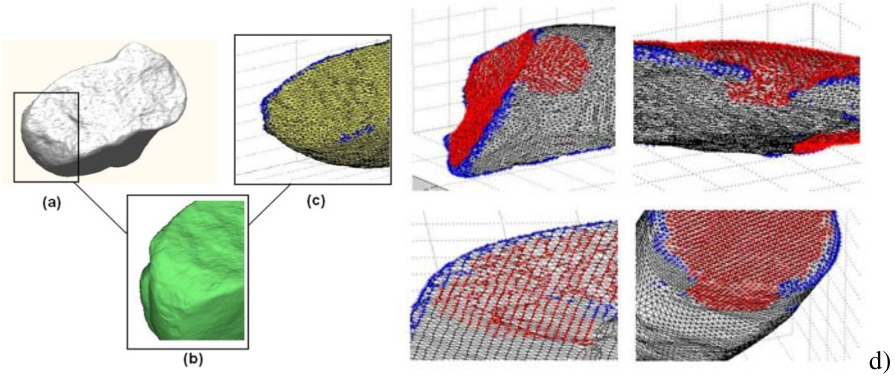


Fig. 2. a) View of the whole object. b) Partial view of the edge segment especially smooth. c) Edge nodes detected by a conventional algorithm. d) Incorrect segmented faces due to the inappropriate detection of the edges.

and usually we could note that the region continues growing, exceeding their real contours. In Figure 2 d) we show segmented faces that are clearly incorrect due to an inappropriate detection of the edges.

The results obtained in the experiments with archaeological objects demonstrate that it is very difficult to implement an effective segmentation algorithm based on edge detection techniques over conventional criteria (i.e. Gaussian curvature). For this reason, it is necessary to use a more robust method.

2 Segmentation Algorithm

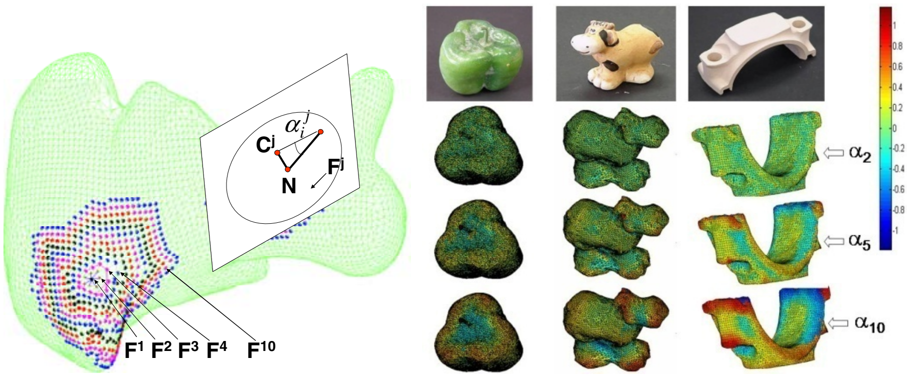
In this section we present a segmentation algorithm which combines edge detection based on CC, edge cycling paths obtaining and region growing. Since Cone Curvature is the geometrical feature we have used, a brief introduction about it follows.

2.1 Cone Curvature Feature

Cone curvature is a geometrical feature originally defined and applied on spherical meshes [13] and lately used for clustering purposes in triangular meshes [14].

Let M_T be a triangular mesh fitted to an object and let N be a vertex of M_T. Note that M_T has been previously regularized and resampled to a fixed number of nodes *h*. We organize the rest of the nodes of the mesh in concentric groups spatially disposed around N. Since this organization resembles the shape of a wave it is called *Modeling Wave (MW)*. Consequently, each of the disjointed subsets is known as *Wave Front (WF)* and the initial node N is called *Focus*. Let us call all the possible MWs that can be generated over T *Modeling Wave Set (MWS)*. Figure 3 a) illustrates the mesh model of an object and several *wave fronts* plotted for two different *focuses*.

Cone Curvature is defined taking into account MWS structure. We define *Cone Curvature* α^j of N∈ M_T, the angle of the cone with vertex N whose surface inscribes the *j*th *Wave Front* of the *Modeling Wave* associated to N.

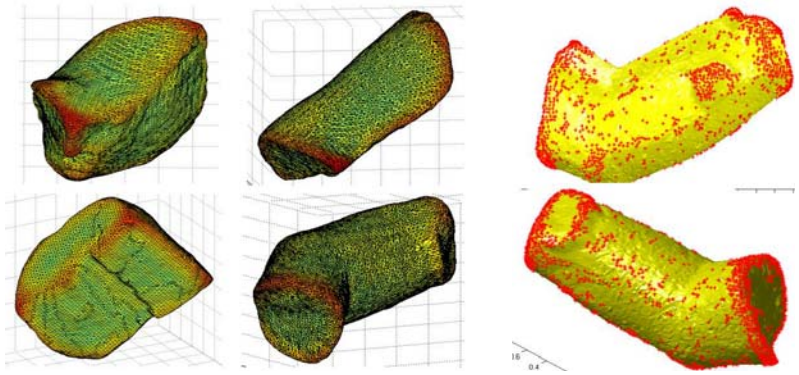


a) b)

Fig. 3. a) *Wave fronts* in two different *focuses* printed over the mesh model and representation of j-th cone curvature in a point of the object. b) Illustration of several CC orders.

The range of CC values is [-π/2, π/2], being the sign assigned taking into account the relative location C^j with respect to M_T, C^j being the barycentre of the *j*th WF. Figure 2 right illustrates this definition.

Note that a set of values $\{\alpha^1, \alpha^2, \alpha^3, \ldots \alpha^q\}$ gives an extended curvature information around N until the *q*th WF, where the word 'curvature' has a non-local meaning. So for each node N a set of *q* values could be used for exploring its surroundings. From now on we will call them *Cone Curvature Levels.* Thus, as the ring increases a new CC level is defined. Thus the Cone Curvatures offer wider information about the surroundings of a point and are a very flexible descriptor because it is possible to select one or more CC levels according to the specifications of the shape to be analyzed. Figure 3 b) illustrates the CC values for a set of orders following a color scale over geometrical models belonging to non-fractured objects.



a) b)

Fig. 4. a) Objects represented with the cone curvature values of its nodes in a colored scale. b) Edge-nodes detected for low (level 4) and high (level 15) CC levels.

In Figure 4 a) we also show a graphical representation of the 5th Cone Curvature calculated on a set of fractured pieces. Note that the regions in red are those which have highest cone curvature values and the regions colored in green are those which have the smaller values. Figure 4 b) presents edge-nodes for several CC values.

2.2 Segmentation Using CC

In fractured objects the segmentation of the original triangular model (M_T) taking into account only one specific CC level might not be achieved. Thus, to define the set of edges in the mesh and to find cycling-edges paths we need to test several CC levels as well as to turn the triangular mesh representation (used in conventional CAD tools and 3D scanner software) into a semi-regular mesh with controlled topology.

Note that if we take a low CC level, we will have discontinuous sets of edge-nodes whereas if we take high CC levels, we might obtain crowded edges. Firstly, in fractured pieces like the ones presented in Figure 1, there exist edges or part of edges which are sharp whereas others are smooth, probably due to erosion effects that the pieces suffer over time. Theoretically the sharp edges can be detected for low CC levels and the smooth ones for medium or, maybe, high CC levels. In practice it doesn't happen in all the cases. Secondly, in both cases using a triangular-patch mesh without a regular topology it is really hard, if not impossible, to find edge-paths in such a geometrical structure. Therefore, in both cases, the extraction of edges, the connection between edges and the search for cycling edge-paths will be inefficient.

In order to control the topological problem, a new model M_E fitted to the object is used. The model M_E has regular 3-connectivity with invariable number of nodes. This canonical model comes from the implosion of a tessellated and semi-regular sphere over the object. Figure 5 a) illustrates the transformation M_T to M_E. This model corresponds to the initial Modeling Wave (MW) topology built on spherical representation models [11]. In the aforementioned topology a node is 3-connected with its neighbors but also is connected, in recursive manner, with the neighbors of the neighbors.

Assuming that the spherical model of the object has been generated, the segmentation algorithm is composed of the following stages.

1.- Calculate the cone curvature values on M_T for a minimum initial level. This value is imposed trough a lineal function taking into account the density of the mesh M_T.

2.- Define the set of edge-node candidates G on M_T by imposing a CC threshold μ. This threshold is calculated through the histogram of CC at the specified level.

3.- Map G into model M_E.

4.- Filter outlier edge-nodes. For instance outlier edge-nodes are nodes which have less than two edge-node neighbors.

5.- Find minimum closed-paths in G trough the controlled topology of M_E. The algorithm is based on a graph-theory recursive search. Generate cycling edge-paths

6.- If there exist open paths in G then take the next CC level an go to step 2

7.- Remap the edges into M_T.

8.- Take an initial random seed and apply a region growing algorithm on M_T.

9.- Save the segment and update the search regions.

10.- Go to 8 until there are no seeds.

Note that, when we have obtained a correct detection of the edges and the cycling boundaries of the regions (step 8), we can employ a region growing algorithm to define the faces of the object. It is well known that region growing is one of the simplest region-based segmentation methods. The region growing method begins taking a random seed on M_T. We check the adjacent nodes and add them to the region if they are not marked as edge-node. Thus the region grows until all nodes of its boundary have some edge-node neighbor. Figure 5b) illustrates results in several steps of the segmentation algorithm.

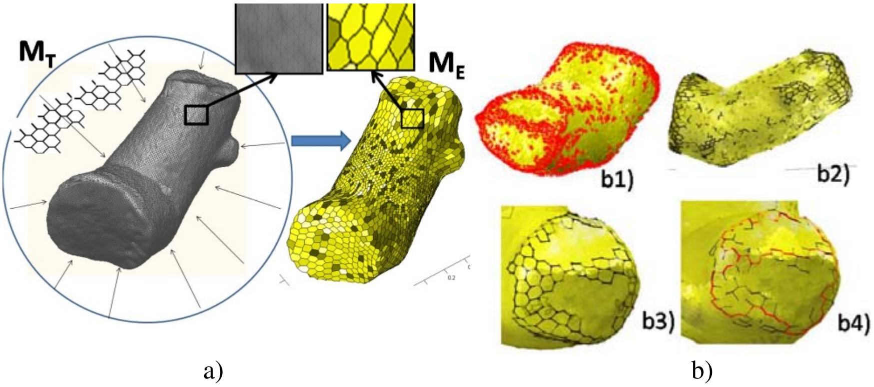


Fig. 5. a) Model transformation. Generating model M_E by implosion of a spherical tessellated mesh into M_T. b1) The set of edge-node candidates on M_T. b2) Edge-node candidates mapped onto model M_E. b3) Result after filtering outlier edge-nodes, b4) Generating cycling paths in M_E and remapping into M_T.

3 Experiments and Future Work

The fragments we are currently segmenting and labeling belong to old sculptures. Our lab, in collaboration with the National Museum of Roman Art of Mérida (Spain) research department, is currently working on a project concerning the digitalization and reconstruction of such sculptural groups dating to the first century B.C. One of the purposes of this project is to solve integration problems of the synthesized models of original pieces in a hypothetical virtual model of the whole sculptural group. In practice, a few real pieces are available. Then, a possible solution is based on, having a priori knowledge of the original sculpture, developing new surface correspondence techniques - which include heuristics and the aid of expert systems - which help to place each part in its original position. In this general proposal, segmentation and labelling of the faces of a fractured piece is a crucial task in making the registering algorithm efficient enough.

Figure 6 a) shows the work developed so far on a sculptural group where only 30 pieces have been recovered. It can be seen the integration of different fractured pieces in the virtual model. Three of them are big pieces that can be easily recognized as body parts. The rest of the pieces are smaller fragments and their identity and position in the original group is currently unknown. Some of these pieces and the results after

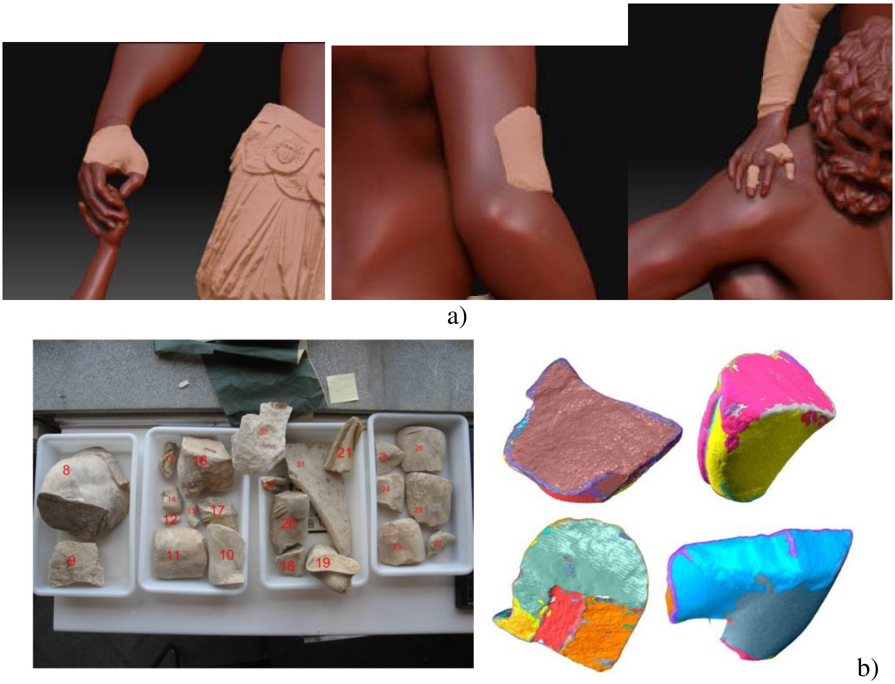


Fig. 6. a) Coupling of several fragments into a virtual model. b)Set of segments belonging to a original sculpture and segmentation results.

segmentation can be seen in Figure 6 b). Beard in mind that the segmented areas do not follows a rigid and usual pattern like in non-manipulated object cases. Here the face contours have irregular shapes. We are currently working on improving the segmentation and labeling of more complex pieces and cases. This is a very difficult problem which needs to be well defined and solved. Therefore, we aspire to solve the problems that we encountered on some of the previous related works. The presented solution should of course be improved and generalized for a wider variety of objects in the archaeological area.

Future works are addressed to develop efficient intelligent algorithms to help archaeologist to reconstruct incomplete 3D puzzles. In this sense, we aim to extend and improve the current semi-automatic solutions and provide an expert system based on fuzzy logic which is able to propose a limited number of solutions which can be evaluated by historians and archaeologist.

4 Conclusions

Standard local feature based approaches are clearly inefficient to make an efficient segmentation in old fractured pieces. This paper presents a new solution in this field by using edge detection algorithms based on the cone–curvature concept. CC allows us to explore the surface from multiple neighborhood levels and to find a compact segmentation solution which characterizes different parts of the piece. A wide experimentation has been carried out on several old pieces belonging to Spanish Museum of Roman Art yielding promising results.

Acknowledgment

This work is supported by DPI2006-14794-C02 and PCI08-0052-1401 projects and with the collaboration of the Spanish Museum of Roman Art.

References

1. Haralick, R.M., Shapiro, L.G.: Image Segmentation Techniques. Computer Vision, Graphics, and Image Processing (January 1985)
2. Pal, N.R., Pal, S.K.: A Review on Image Segmentation Techniques. Pattern Recognition (1993)
3. Liu, H.K.: Two and Three Dimensional Boundary Detection. Computer Graphics and Image Processing (April 1977)
4. Herman, G.T., Liu, H.K.: Dynamic Boundary Surface Detection. Computer Graphics and Image Processing (1978)
5. Zucker, S.W., Hummel, R.A.: An Optimal Three- dimensional Edge Operator. Technical report, McGil University, Toronto, Ontario, Canada (April 1979)
6. Huang, Q.-X.: Reassembling Fractured Objects by Geometric Matching. ACM Transactions on Graphics (TOG) archive 25(3) (July 2006)
7. Pottman, H., Wallner, J., Huang, Q.-X., Yang, Y.: Integral invariants for robust geometry processing. Computer Aided Geometric Design 26, 37–60 (2009)
8. Bellon, O., Silva, L.: New improvements to range image segmentation by edge detection. IEEE Signal Processing Letters 9(2), 43–45 (2002)
9. Gotardo, P., Bellon, O., Boyer, K., Silva, L.: Range Image Segmentation Into Planar and Quadric Surfaces Using an Improved Robust Estimator and Genetic Algorithm. IEEE Transactions on Systems, Man and Cybernetics. Part B, Cybernetics 34(6), 2303–2316 (2004)
10. Papaioannou, G., Karabassi, E.-A., Theoharis, T.: Virtual Archaeologist, Assembling the Past. IEEE, Computer Graphics and Applications 21, 53–59 (2001)
11. Papaioannou, G., Karabassi, E.-A., Theoharis, T.: Reconstruction of three-dimensional objects through matching of their parts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 114–124 (2002)
12. Papaioannou, G., Karabassi, E.-A.: Automatic Reconstruction of Archaeological Finds – A Graphics Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 114–124 (2002)
13. Adán, A., Adán, M.: A Flexible Similarity Measure for 3D Shapes Recognition. IEEE Transactions on Pattern Analysis and Machine Inteligence 26(11) (November 2004)
14. Adán, A., Adán, M., Salamanca, S., Merchán, P.: Using Non Local Features For 3D Shape Grouping. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 644–653. Springer, Heidelberg (2008)

Segmentation in 2D and 3D Image Using Tissue-Like P System

Hepzibah A. Christinal[1,2], Daniel Díaz-Pernil[1], and Pedro Real Jurado[1]

[1] Research Group on Computational Topology and Applied Mathematics
University of Sevilla
Avda. Reina Mercedes s/n, 41012, Sevilla, Spain
[2] Karunya University
Coimbatore, Tamilnadu, India
{hepzi,sbdani,real}@us.es

Abstract. Membrane Computing is a biologically inspired computational model. Its devices are called P systems and they perform computations by applying a finite set of rules in a synchronous, maximally parallel way. In this paper, we open a new research line: P systems are used in Computational Topology within the context of the Digital Image. We choose for this a variant of P systems, called *tissue-like P systems*, to obtain in a general maximally parallel manner the segmentation of 2D and 3D images in a constant number of steps. Finally, we use a software called *Tissue Simulator* to check these systems with some examples.

1 Introduction

Natural Computing studies new computational paradigms inspired from Nature. It abstracts the way in which Nature "computes", conceiving new computing models. There are several fields in Natural Computing that are now well established as are Genetic Algorithms ([8]), Neural Networks ([10]), DNA-based molecular computing ([1]).

Membrane Computing is a theoretical model of computation inspired by the structure and functioning of cells as living organisms able to process and generate information. The computational devices in Membrane Computing are called *P systems* [15]. Roughly speaking, a P system consists of a membrane structure, in the compartments of which one places multisets of objects which evolve according to given rules. In the most extended model, the rules are applied in a synchronous non-deterministic maximally parallel manner, but some other semantics are being explored.

According to their architecture, these models can be split into two sets: cell-like P systems and tissue-like P systems [19]. In the first systems, membranes are hierarchically arranged in a tree-like structure. The inspiration for such architecture is the set of vesicles inside the cell. All of them perform their biological processes in parallel and life is the consequence of the harmonious conjunction

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 169–176, 2009.

of such processes. This paper is devoted to the second approach: tissue-like P systems.

Segmentation in computer vision (see [9]), refers to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

There exists different techniques to segment an image. Few techniques are Clustering methods [7], Histogram-based methods [12], Watershed transformation methods [23], Graph partitioning methods [22].

Some of the practical applications of image segmentation are like Medical Imaging [7], Study of Anatomical Structure, Locate objects in Satellite Images (roads, forests, etc.) [11], Face Recognition [21].

J. Chao and J. Nakayama connected in [4] Natural Computing and Algebraic Topology using Neural Networks (extended Kohonen mapping). We use for the first time, the power and efficiency of a variant of P systems called tissue-like P systems(see [5]) to segment the image in 2D.

The paper is structured as follows: in the next section we present the definition of basic tissue like P systems with input and show an example to understand how these systems work. In section 3, we design a family of systems for edge-based segmentation in 2D image. After, we check our model using a software called tissue simulator with two images very easy. At the end of this section, we introduce a family of tissue-like P systems to obtain an edge-based segmentation of 3D images. Finally, some conclusions and future work are given in the last section.

2 Description of a Model of Membranes

Membrane computing models was first presented by Martín–Vide *et al.* in [13] and it has two biological inspirations (see [14]): intercellular communication and cooperation between neurons, but in this paper we work with a variant presented in [19] *with cell division* and the system which is presented by Díaz-Pernil presented in [6] a formalization of *Tissue-like P systems* (without cellular division). The common mathematical model of these two mechanisms is a network of processors dealing with symbols and communicating these symbols along channels specified in advance.

The main features of this model, from the computational point of view, are that cells have not polarizations (the contrary holds in the cell-like model of P systems, see [16]) and the membrane structure is a general graph.

Formally, a *tissue-like P system* with input of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, \ldots, w_q, \mathcal{R}, i_\Pi, o_\Pi),$$

where

1. Γ is a finite *alphabet*, whose symbols will be called *objects*, $\Sigma(\subset \Gamma)$ is the input alphabet, $\mathcal{E} \subseteq \Gamma$ (the objects in the environment),
2. $w_1, \ldots, w_q$ are strings over Γ representing the multisets of objects associated with the cells at the initial configuration,
3. $\mathcal{R}$ is a finite set of communication rules of the following form: $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \ldots, q\}, i \neq j$, $u, v \in \Gamma^*$,
4. $i_\Pi, o_\Pi \in \{0, 1, 2, \ldots, q\}$.

A tissue-like P system of degree $q \geq 1$ can be seen as a set of q cells (each one consisting of an elementary membrane) labelled by $1, 2, \ldots, q$. We will use 0 to refer to the label of the environment, i_Π denotes the input region and o_Π denotes the output region (which can be the region inside a cell or the environment).

The strings $w_1, \ldots, w_q$ describe the multisets of objects placed in the q cells of the system. We interpret that $\mathcal{E} \subseteq \Gamma$ is the set of objects placed in the environment, each one of them available in an arbitrary large amount of copies.

The communication rule $(i, u/v, j)$ can be applied over two cells labelled by i and j such that u is contained in cell i and v is contained in cell j. The application of this rule means that the objects of the multisets represented by u and v are interchanged between the two cells. Note that if either $i = 0$ or $j = 0$ then the objects are interchanged between a cell and the environment.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered). In one step, each object in a membrane can only be used for one rule (non-deterministically chosen when there are several possibilities), but any object which can participate in a rule of any form must do it, i.e, in each step we apply a maximal set of rules.

Now, to understand how we can obtained a computation of one of these P systems we present an example of them:

Consider us the tissue-like P system $\Pi' = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$ where

1. $\Gamma = a, b, c, d, e$, $\Sigma = \emptyset$, $\mathcal{E} = a, b, e$,
2. $w_1 = a^3\, e$, $w_2 = b^2\, c\, d$,
3. $\mathcal{R} = \{(1, a/b, 2), (2, c/b^2, 0), (2, d/e^2, 0), (1, e/\lambda, 0)\}$,
4. $i_\Pi = 1$ and $o_\Pi = 0$.

We can observe the initial configuration of this system in the figure 1 (a). We have four rules to apply, and after applying the rules the next configuration is

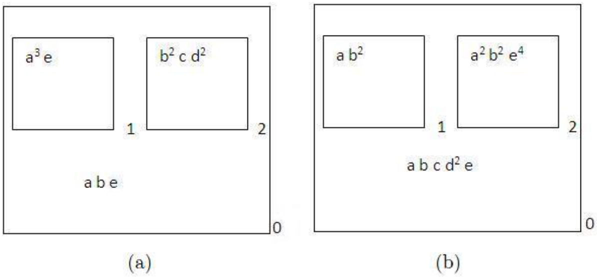


Fig. 1. (a) Initial Configuration of system Π' (b) Following Configuration of Π'

shown in (b), the system apply one time each one of them. If reader looks at the elements in the environment one can observe the number of the copies of the elements a, b, e always are one, because they are the objects that appear in the environment initially (we have an arbitrary large amount of copies of them), but d has two copies because it is not an initial element of the environment. Usually, the elements of the environment are not described in the system to better understanding of the configurations of this.

3 Segmenting Digital Images in Constant Time

In this section, we segment images based on edge-based segmentation. Edge-based segmentation finds boundaries of regions which are sufficiently different from each other. We define two family of tissue-like P systems for edge-based segmentation, one of them to segment 2D images and after, we adapt these systems to segment 3D images.

3.1 A Family of Tissue-Like P Systems for a 2D Segmentation

We can divide the image in multiple pixels forming a network of points of $\mathbb{N}^2$. Let $\mathcal{C} \subseteq \mathbb{N}$ be the set of all colors in the given 2D image and they are in a certain order. Moreover, we will suppose each pixel is associated with a color of the image. Then we can codify the pixel (i,j) with associated color $a \in \mathcal{C}$ by the object a_{ij}.

The following question is the adjacency problem. We have decided to use in this paper the 4-adjacency [2,3].

In this point, we want to find the border cells of the different color regions present in the image. Then, for each image with $n \times m$ pixels ($n, m \in \mathbb{N}$) we will construct a tissue-like P system whose input is given by the objects a_{ij} codifying a pixel, with $a \in \mathcal{C}$. The output of the system is given by the objects that appear in the output cell when it stops.

So, we can define a family of tissue-like P systems to do the edge-based segmentation to 2D images. For each $n, m \in \mathbb{N}$ we consider the tissue-like P system $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$, defined as follows:

(a) $\Gamma = \Sigma \cup \{\bar{a}_{ij} : 1 \leq i \leq n,\ 1 \leq j \leq m\} \cup \{A_{ij} : 1 \leq i \leq n,\ 1 \leq j \leq m,\ A \in \mathcal{C}\}$, $\Sigma = \{a_{ij} : a \in \mathcal{C},\ 1 \leq i \leq n,\ 1 \leq j \leq m\}$, $\mathcal{E} = \Gamma - \Sigma$,
(b) $w_1 = w_2 = \emptyset$,
(c) R is the following set of communication rules:

1. $(1, a_{ij} b_{kl} / \bar{a}_{ij} A_{ij} b_{kl}, 0)$, for $a, b \in \mathcal{C},\ a < b,\ 1 \leq i, k \leq n,\ 1 \leq j, l \leq m$ and (i, j), (k, l) adjacents.
   These rules are used when image has two adjacent pixels with different associated colors(border pixels), and the pixel with less associated color is marked and system brings from the environment an object representing this marked pixel(edge pixel).

2. $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i+1j+1}b_{i+1j} \,/\, \bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i+1j+1}b_{i+1j}, 0)$ for $a, b \in \mathcal{C}, a < b$, $1 \leq i \leq n-1,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{i-1j}\bar{a}_{i-1j+1}b_{ij+1} \,/\, \bar{a}_{ij}\bar{a}_{i-1j}A_{i-1j}\bar{a}_{i-1j+1}b_{ij+1}, 0)$ for $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{ij+1}\bar{a}_{i-1j+1}b_{i-1j} \,/\, \bar{a}_{ij}\bar{a}_{ij+1}A_{ij+1}\bar{a}_{i-1j+1}b_{i-1j}, 0)$ for $a, b \in \mathcal{C}, a < b, 2 \leq i \leq n,\ 1 \leq j \leq m-1$.
   $(1, \bar{a}_{ij}a_{i+1j}\bar{a}_{i+1j+1}b_{ij+1} \,/\, \bar{a}_{ij}\bar{a}_{i+1j}A_{i+1j}\bar{a}_{i+1j+1}b_{ij+1}, 0)$ for $a, b \in \mathcal{C}, a < b, 1 \leq i \leq n-1,\ 1 \leq j \leq m-1$.
   The rules mark with a bar the pixels which are adjacent to two same color pixels and which were marked before, but with the condition that the marked objects are adjacent to an other pixel with a different color. Moreover, an edge object representing the last marked pixel is brought from the environment.

3. $(1, A_{ij}/\lambda, 2)$, for $1 \leq i \leq n,\ 1 \leq j \leq m$.
   This rule is used to send the edge pixels to the output cell.

d) $i_\Pi = 1$, $o_\Pi = 2$.

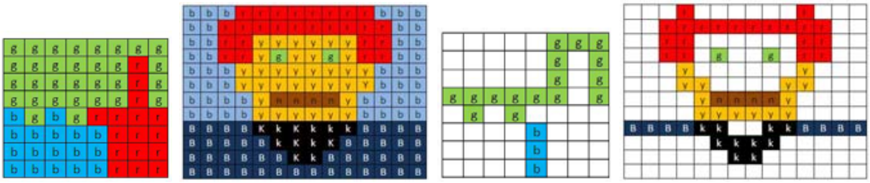
An overview of the Computation: Input objects a_{ij} codifying the colored pixels from an 2D image appear in the input cell and with them the system begins to work. Rules of type 1, in a parallel manner, identify the border pixels and bring the edge pixels from the environment. These rules need 4 steps to mark all the border pixels. From the second step, the rules of type 2 can be used with the first rules at the same time. So, in other 4 steps we can bring from the environment the edge pixels adjacent to two border pixels as we explain above. System can apply the first two types of rules simultaneously in some configurations, but it always applies the same number of these two types of rules because this number is given by edge pixels(we consider 4-adjacency). Finally, the third type of rules are applied in the following step on edge pixels appear in the cell. So, with one step more we will have all the edge pixels in the output cells. Thus, we need only 9 steps to obtain an edge-based segmentation for an $n \times m$ image. Then, we can conclude the problem of edge-segmentation in 2D images is resolved in this paper in a constant time respect to the number of steps of any computation.

3.2 Checking This Family of Systems with Tissue Simulator

We have used a software called *tissue-simulator* (See section 2) introduced by Borrego-Ropero et al. in [20]. We have simulated our family of systems to segment 2D images with this software. Finally, we have introduced as instances of our system the examples that appear in 3 (a) and, in a constant number of steps we have obtained a codifying of the edge-segmentation (that appear in 3 (b)) of the examples introduced before.

We consider, in a first case an 8×8 image, and the order of the colors used in this image is the following: green, blue and red. In a second case we work with an image of size 12×14. In this example, we take the colors in the following order: Red, green, brown, orange, black, blue and light blue.

Fig. 2. Two images about Tissue Simulator



g	g	g	g	g	g	g	g
g	g	g	g	g	g	r	g
g	g	g	g	g	g	r	g
g	g	g	g	g	g	r	g
b	g	b	g	r	r	r	r
b	b	b	b	b	r	r	r
b	b	b	b	b	r	r	r
b	b	b	b	b	r	r	r

b	b	b	r	r	r	r	r	r	r	r	b	b	b
b	b	r	r	r	r	r	r	r	r	r	r	b	b
b	b	r	r	y	y	y	y	y	y	r	r	b	b
b	b	r	r	y	g	y	y	g	y	r	r	b	b
b	b	b	y	y	y	y	y	y	y	y	b	b	b
b	b	b	y	y	y	y	y	y	y	Y	b	b	b
b	b	b	b	y	n	n	n	n	y	b	b	b	b
b	b	b	b	y	y	y	y	y	y	b	b	b	b
B	B	B	B	K	k	K	k	k	k	B	B	B	B
B	B	B	B	B	k	K	k	K	B	B	B	B	B
B	B	B	B	B	B	K	k	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B	B

(a)

					g	g	g
					g		g
					g		g
g	g	g	g	g	g		g
	g		g				
				b			
				b			
				b			

			r							r			
		r	r	r	r	r	r	r	r	r	r		
		r	r							r	r		
		r	r		g			g		r	r		
			y							y			
			y	y					y	y			
				y	n	n	n	n	y				
				y	y	y	y	y	y				
B	B	B	B	k	k			k	k	B	B	B	B
					k	k	k	k					
						k	k						

(b)

Fig. 3. (a) Input images (b) Edge-segmentation of images
r-red; b-blue; B-dark blue; g-green; y-yellow; n-brown; k-black; blank-white.

3.3 Segmenting 3D Images

The following step to consists to extend our models to work with 3D images. Now, the input dates are voxels ($(i, j, k) \in \mathbb{N}^3$) that the are codifying by the elements a_{ijk}, with $a \in \mathcal{C}$.We use here 26-adjacency relationship between voxels. Then, we can define a family of tissue-like P systems. For each $n, m \in \mathbb{N}$ we consider the $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, w_2, \mathcal{R}, i_\Pi, o_\Pi)$ to do an edge-based segmentation to a 3D image as follows

(a) $\Gamma = \Sigma \cup \{A_{ijk} \ : 1 \le i \le n,\ 1 \le j \le m, 1 \le k \le l,\ a \in \mathcal{C}\}$, $\Sigma = \{a_{ijk} : a \in \mathcal{C},\ 1 \le i \le n,\ 1 \le j \le m, 1 \le k \le l\}$, $\mathcal{E} = \Gamma - \Sigma$,

(b) $w_1 = w_2 = \emptyset$,

(c) R is the following set of rules:

1. $(1, a_{i_1 j_1 k_1} b_{i_2 j_2 k_2} / A_{i_1 j_1 k_1} b_{i_2 j_2 k_2}, 0)$, for $1 \le i_1, i_2 \le n$, $1 \le j_1, j_2 \le m$, $1 \le k_1, k_2 \le l$, (i_1, j_1, k_1) and (i_2, j_2, k_2) adjacents voxels and finally, $a, b \in \mathcal{C}$ with $a < b$.

   These rules are used when image has two adjacent border voxels. Then, system brings from the environment an object representing the voxel with less associated color (edge voxel).

2. $(1, A_{ijk}/\lambda, 2)$, for $1 \le i \le n,\ 1 \le j \le m, 1 \le k \le l$.

   These rules are used to send the edge voxels to the output cell.

d) $i_\Pi = 1$, $o_\Pi = 2$.

An overview of the Computation: This computation would be very similar if we consider an 26-adjacency in 3D. Rules of type 1 identify the border pixels and bring the edge pixels from the environment. These rules need as much 26 steps for this. Finally, the second type of rules are applied in the following step and send the edge pixels to the output cell. So, we need again a constant amount of steps to resolve the edge-segmentation problem in 3D.

4 Conclusions and Future Work

It is shown in this paper, if we consider a 4-adjacency, a segmentation of a 2D image can be given in a constant number of steps using tissue-like P systems.

With this paper new research lines have been opened. We can work in some of them directly, define new systems to obtain other homological informations (spanning trees, homology gradient vector field, representative cocycles of cohomology generators, etc) for 2D or 3D images. But, other lines need more time and deep research work, as are: to develop an efficient sequential software using these techniques, to develop an efficient software working with a cluster. Moreover, both of them could be applied in different areas as are: medical imaging, locate objects in satellite in satellite images, etc.

Acknowledgement

The first author acknowledge the support of the project ”Computational Topology and Applied Mathematics” PAICYT research project FQM-296. The second author acknowledge the support of the project TIN2006-13425 of the Ministerio de Educación y Ciencia of Spain, and the support of the Project of Excellence of the Junta de Andalucía, grant P08-TIC-04200. The third author acknowledge the support of the project MTM2006-03722 of the Ministerio español de Educación y Ciencia and the project PO6-TIC-02268 of Excellence of Junta de Andalucía.

References

1. Adleman, L.M.: Molecular computations of solutions to combinatorial problems. Science 226, 1021–1024 (1994)
2. Rosenfeld, A.: Digital topology. American Mathematical Monthly 86, 621–630 (1979)
3. Rosenfeld, A.: Connectivity in Digital pictures. Journal for Association of Computing Machinery 17(1), 146–160 (1970)
4. Chao, J., Nakayama, J.: Cubical Singular Simples Model for 3D Objects and Fast Computation of Homology Groups. In: Proceedings of ICPR 1996 IEEE, pp. 190–194 (1996)
5. Díaz-Pernil, D., Gutiérrez, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: A uniform family of tissue P systems with cell division solving 3-COL in a linear time. Theoretical Computer Science 404, 76–87 (2008)
6. Díaz-Pernil, D., Pérez-Jiménez, M.J., Romero, A.: Efficient simulation of tissue-like P systems by transition cell-like P systems. Natural Computing, http://dx.doi.org/10.1007/s11047-008-9102-z

7. Wang, D., Lu*, H., Zhang, J., Liang, J.Z.: A Knowledge-Based Fuzzy Clustering Method with Adaptation Penalty for Bone Segmentation of CT images. In: Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pp. 6488–6491 (2005)
8. Holland, J.H.: Adaptation in Natural and Artificial Systems (1975)
9. Shapiro, L.G., Stockman, G.C.: Computer Vision (2001)
10. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943)
11. Sharmay, O., Miocz, D., Antony, F.: Polygon Feature Extraction from Satellite imagery based on color image segmentation and medial axis. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII, 235–240 (2008)
12. Tobias, O.J., Seara, R.: Image Segmentation by Histogram Thresholding Using Fuzzy Sets. IEEE Transactions on Image Processing 11(12), 1457–1465 (2002)
13. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez-Patón, A.: A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. In: Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, pp. 290–299. Springer, Heidelberg (2002)
14. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez Patón, A.: Tissue P systems. Theoretical Computer Science 296, 295–326 (2003)
15. Păun, G.: Computing with membranes. Journal of Computer and System Sciences 61, 108–143 (2000)
16. Păun, G.: Membrane Computing: An Introduction. Springer, Heidelberg (2002)
17. Păun, A., Păun, G.: The power of communication: P systems with symport/antiport. New Generation Computing 20(3), 295–305 (2002)
18. Păun, G.: Computing with Membranes: Attacking **NP**–complete Problems. In: Unconventional Models of Computation, UMC'2K, pp. 94–115. Springer, Heidelberg (2000)
19. Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Tissue P System with cell division. In Second Brainstorming Week on Membrane Computing, Sevilla, Report RGNC 01/2004, pp. 380–386 (2004)
20. http://www.tissuesimulator.es.kz
21. Kim, S.-H., Kim, H.-G., Tchah, K.-H.: Object oriented face detection using colour transformation and range segmentation. IEEE Electronics Letters 34, 979–980 (1998)
22. Xiaojing, Y., Ning, S., George, Z.: A narrow band graph partitioning method for skin lesion segmentation. Elsevier Science Pattern Recognition 42(6), 1017–1028 (2009)
23. Yazid, H., Arof, H.: Image Segmentation using Watershed Transformation for Facial Expression Recognition. IFMBE Proceedings, 4th Kuala Lumpur International Conference on Biomedical Engineering 21, 575–578 (2008)

Dynamic Image Segmentation Method Using Hierarchical Clustering

Jorge Galbiati[1], Héctor Allende[2,3,⋆], and Carlos Becerra[4]

[1] Pontificia Universidad Católica de Valparaíso, Chile, Department of Statistics
jorge.galbiati@ucv.cl
[2] Universidad Técnica Federico Santa María, Chile, Department of Informatics
hallende@inf.utfsm.cl
[3] Universidad Adolfo Ibáñez, Chile, Science and Ingeneering Faculty
[4] Universidad de Valparaíso, Chile, Department of Computer Science
carlos.becerra@uv.cl

Abstract. In this paper we explore the use of the cluster analysis in segmentation problems, that is, identifying image points with an indication of the region or class they belong to. The proposed algorithm uses the well known agglomerative hierarchical cluster analysis algorithm in order to form clusters of pixels, but modified so as to cope with the high dimensionality of the problem. The results of different stages of the algorithm are saved, thus retaining a collection of segmented images ordered by degree of segmentation. This allows the user to view the whole collection and choose the one that suits him best for his particular application.

Keywords: Segmentation analysis, growing region, clustering methods.

1 Introduction

Image segmentation is one of the primary steps in image analysis for object identification. The main aim is to recognize homogeneous regions within an image as distinct objects. It consists of partitioning an image in regions that are supposed to represent different objects within it. Some segmentation methods are automatic, or non-supervised, they do not need human interaction; others are semiautomatic or supervised.

One of the simplest technique used for segmentation is thresholding, in which pixels whose intensity exceeds a threshold value defined by the analyst are said to belong to one region, while those that do not, belong to the other.

A semiautomatic segmentation methods is region growing, that starts off with a group of pixels defined by the analyst as seeds, then other neighborhood pixels are added if they have similar characteristics, according to some criterion. This algorithm is improved with the introduction of a bayesian approach, Pan and Lu [8], by means of a homogeneity criteria of neighbouring pixels.

While image segmentation consists of partitioning an image in homogeneous regions, edge detection is the identification of the lines that define the borders

⋆ This work was supported by Fondecyt 107220 research grant.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 177–184, 2009.

between regions determined by a segmentation process. Image segmentation and edge detection are both dual problems, in the sense that solving one provides the solution to the other. Some segmentation methods are based on edge detection. That is the case of morphological watershed methods. They consist on calculating the gradients of an image, and make them to resemble mountains, while simulating a water flood, the water level raising progressively. As the water level increases, it determines thinner zones on the top of the mountains, which are defined as edges, and the zones contained within these edges are segments. Frucci et al. [4] developed a segmentation method based on watersheds using resolution pyramids, formed by a set of images of decreasing resolution, under the premise that the most relevant aspects of an image are perceived even at low resolution.

The use of wavelets has spread as a segmentation method. Wavelets can model the behavior pattern of the pixels in the image. The difference with the Fourier transformation is that these are not periodic. Perhaps the most widely used wavelet transformation in images is the Gabor wavelet, the product of a sinusoid and a gaussean. Wang et al. [10] introduced a texture segmentation method on the basis of Gabor wavelets. It is used for the extraction of texture characteristics.

Image segmentation by hierarchic clusters uses the classic hierarchic cluster analysis methods that groups the nearest clusters at each stage. Martinez-Uso et al. [7] segment an image by means of a hierarchic process that maximizes a measurement that represents a perceptual decision. It can be also used for multispectral images. Another clustering method is k-means, that iteratively moves elements to the cluster whose centroid is the closest, the process ends when no elements change places. Fukui et al. [5] apply hierarchic clustering for the detection of objects, in particular, face detection. They use k- means clustering with color space features. Allende and Galbiati [2] developed a method for edge detection in contaminated images, based on agglomerative hierarchical clusters, by performing a cluster analysis on a 3×3 pixel moving window; more than one cluster denotes the presence of an edge. A segmentation method based on cluster analysis is shown in Allende et al. [3]. It runs a 3×3 neighbourhood window along the image, performing a cluster analysis each time and deciding whether the central pixel belongs to one of the clusters as the surrounding pixels. If it does, it assigns it to the corresponding cluster. If not, it creates a new cluster. At the end, each cluster is considered a segment.

Image segmentation is usually combined with other image processing methods, like image image enhancement and restoration, which are usually performed before segmenting; and like feature extraction, object recognition and classification, and texture analysis, for which segmentation is a previous step.

2 Method

The agglomerative clustering algorithm starts defining each element as one cluster, so at the starting point we have as many clusters as the number of elements. Then the distances between elements are computed, forming a symmetric $N \times N$

distance matrix, where N is the number of elements. Distances like a Minkowski type distance are frequently used,

$$d(\underline{x}, \underline{y}) = \Big(\sum_{i=1}^{p} |x_i - y_i|^k \Big)^{1/k}$$

where $\underline{x}$ and $\underline{y}$ are p-dimensional vector elements, and k is any positive real number. $k = 1$ gives the "city block" distance, $k = 2$ corresponds to the euclidean distance.

Then we need to define a distance measure between clusters, as a function of the distance between elements. A convenient distance measure is the average of all the distances between pairs of elements, one from each cluster. Other distances are, the smallest distance between pairs of elements, or the largest distance, or the distance between the averages of the elements of both clusters.

The second step consists of finding the smallest distance, and joining the corresponding two clusters to form a new cluster with two elements. The distances from this new cluster to the remaining clusters are then computed, and the distances from the original two clusters that merged are eliminated from the set of distances, so we have a new $(N-1) \times (N-1)$ distance matrix. The previous procedure is repeated until we have one single cluster containing all the elements. At each stage, the distance at which two clusters are merged increases, thus building clusters in increasing degree of heterogeneity. A record is kept of all the stages, the way they join to form larger clusters and their merging distances, which form an increasing sequence, and can be represented by a graph called dendogram, which illustrates the way the hierarchical procedure developed.

As can be noticed, the complexity of the problem grows fast as the number of elements to be clustered increases. In fact, in the case of N elements, the number of distances, and consequently the number of steps, is $\frac{1}{2}N \cdot (N-1)$.

This procedure can be applied to images, to develop a segmentation method, where each pixel is a vector element. In the case of color RGB images, the vectors are three dimensional, the coordinates representing the Red, Green and Blue intensities. In monochrome images, the elements are scalars, and the Minkowski distance turns up to be equal to the absolute value difference.

The algorithm becomes particularly complex in the case of images. An image consisting of n columns and m rows has $npix = n \cdot m$ pixels, so the number of distances is $\frac{1}{2}n \cdot m \cdot (n \cdot m - 1) = \frac{1}{2}n^2 \cdot m^2 - n \cdot m$. Suppose a 1000×1000 image, which is not too large, the number of distances is approximately $5 \cdot 10^{11}$, which would make the task of forming clusters almost impossible. But, unlike general situations where cluster analysis is applied, in images the pixels to be clustered are elements which have an order. And the cluster forming procedure should consider this fact, so as to merge only adjacent clusters. As a consequence, at the initial step the distances to be considered are the distances between each pixel and two of its neighbors, the one at its left and the one beneath. That makes the number of initial distances to be scanned at the first step equal to $ndist = 2m \cdot m - n - m$. In the example where $n = m = 1000$, this number is approximately equal to 2×10^6, considerably smaller that in the general case.

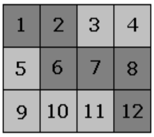


Fig. 1. Segmentation process

Figure 1 illustrates how the algorithm works for an image with three clusters. The pixels that are to be compared are the following: First row, 1-2, 1-5, 2-3, 2-6, 3-4, 3-7, and 4-8; second row, 5-6, 5-9, 6-7, 6-10, 7-8, 7-11, and 8-12; third row, 9-10, 10-11, and 11-12. There are two light gray clusters, but although they are the same colour, their pixels are never going to be compared, as both clusters are not adjacent. Thus, although the same, the algorithm considers them as different clusters.

The algorithm uses three arrays of size $npix$: $Frec(s)$, $Avg(s)$ and $Pos(p)$. $Frec(s)$ keeps the frequency, or number of pixels, of the cluster or segment s; $Avg(s)$ has the average value of the intensity of each color of all pixels within segment s; $Pos(p)$ contains a label assigned to pixel p, indicating the segment to which it belongs and a label giving it a number within the segment. The algorithm also uses three arrays of size $ndist$: $dist(d)$, $e(d)$ and $f(d)$, where $dist(d)$ keeps the d^{th} distance value between segments $e(d)$ and $f(d)$.

At each step, after the smallest distance is obtained, say $dist(d_0)$, the two segments $e(d_0)$ and $f(d_0)$ are merged into one, which retains the label $e(d_0)$. The frequency of the new segment $e(d_0)$ is the sum of the two frequencies, while the frequency of the $f(d_0)$ is set to zero. The average value of this new segment is obtained as a weighted average of the averages of the two concurrent segments, and the average of $f(d_0)$ is now zero. The labels in $Pos(p)$ are updated for each pixel p that belonged to segment $f(d_0)$.

Finally, the distances from the new segment to all its adjacent segments are updated. The adjacent segments are easily identified, considering all pairs $(e(d), f(d))$, where one of the two numbers is the label of any one of the two segments that merged; the other corresponds to an adjacent segment. To start, every pixel is defined as a segment, numbered from 1 to $npix$.

The iterations are carried on until the merging distance reaches a threshold value $thrs$ given initially as an input, an integer in the interval (0,255). With this threshold value the degree of the segmentation can be controlled. Small values will give finer segmentations, while higher values will give coarser segmentation, with a small number of large segments.

The sequence of segment union distances follows the same pattern, independent of the threshold value. The difference is that if we choose two segmentations with different threshold values, say $T1$ and $T2$, with $T1 < T2$, then the sequence of steps followed to reach $T1$ is the same as the initial part of the sequence of steps followed to reach $T2$.

We can take advantage of this fact in order to obtain a series of segmentations, with different degrees of detail, instead of just one. In this way, after the image is processed, we can sequentially visualize all the segmented images, each with a different level of segmentation, from the finest, equivalent to the original image, to the coarsest level, which corresponds to a one color rectangle, the average color. To do this, the dimensionality of the arrays has to be increased, but it is not necessary to save each entire image for every segmentation level, we only have to save the changes made from one step to another, and this reduces the amount of memory required.

The method presented here differs from seeded region growing because here, the algorithm starts merging pixels which are closest together, according to the distance being used, while in the other method the merging starts with pixels which are adjacent to seed pixels that were previously determined by the user.

3 Results

In the experimental results presented in this paper, the distance used was the city block distance. The distance between clusters is the sum of the absolute differences between the average of each colour, in both clusters. In each case we saved 24 segmented images, with threshold values for the maximum merging distance ranging from the minimum to the maximum observed distances in each case, at intervals of one 23rd of the range between them. In Figures 2 to 6 we show a few of the results, indicating the threshold value in each case. Some of the images were taken from the Berkeley Data Set for Segmentation and Benchmark [6].

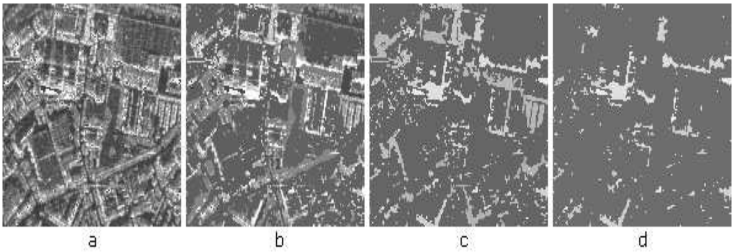


Fig. 2. City air photograph. (a) Original image. Segmentation threshold values: (b) 40, (c) 80, (d) 96.

4 Discussion

The method involves a large amount of computing, that makes it relatively slow, compared to some other methods. Figure 5 is a 131 × 135 image, with a total of 17685 pixels, that makes 35104 initial distances. It took 6 minutes and 28 seconds to process with a Visual Basic program, running on an Intel Duo Core processor at 3.0 GHz, with 3.25 GB Ram. The output was a series of 15 segmented images, four of which are shown here. An optimized computer program would contribute to make it work faster. It is also memory consuming, but a careful management of the stored information can help to reduce the memory consumption.

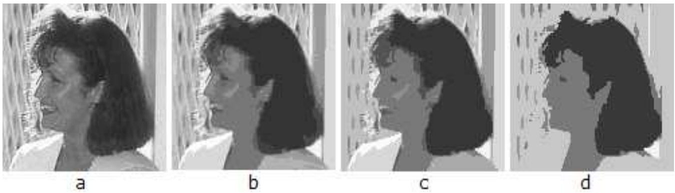


Fig. 3. Woman´s face 1. (a) Original image. Segmentation threshold values: (b) 21, (c) 28, (d) 50.

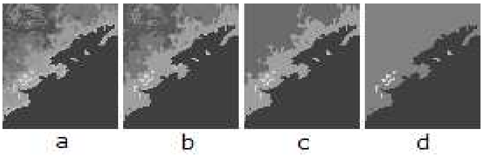


Fig. 4. Air photograph of coast. (a) Original image. Segmentation threshold values: (b) 27, (c) 37, (d) 59.

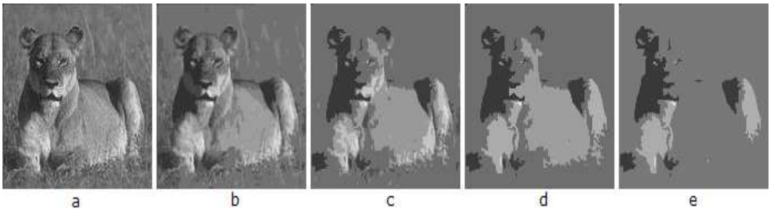


Fig. 5. Female lion. (a) Original image. Segmentation threshold values: (b) 19, (c) 32, (d) 52, (e) 58.

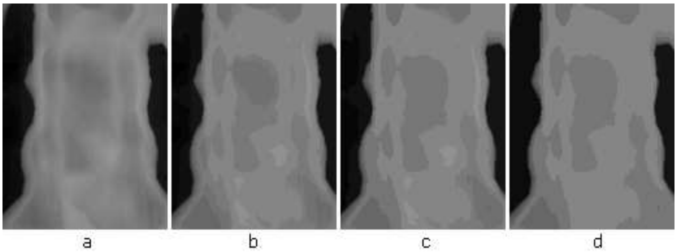


Fig. 6. Vertebrae X-ray. (a) Original image. Segmentation threshold values: (b) 9, (c) 12, (d) 15.

The particular way that the segments are merged together, that is, only considering adjacent segments, results in the fact that sometimes two disjoint segments look very similar in color, but remain as separate segments, while others, that are not as similar, are merged earlier. This is correct, because if two segments remain like that, it is because they are not adjacent. In the image, that means that they are separate, and even if they look as if they are similar in color, they do represent different objects. That can be clearly appreciated in the lion´s leg and body, in Figure 5.

If the image is too noisy, the method does not give a good result. But applying a previous smoothing, like, for example, a median smoothing, or the method shown in [1] the quality of the result is significantly increased.

5 Conclusions and Future Studies

In case we want one particular segmentation we have to set a threshold value at the start of the segmentation algorithm. If we want the algorithm to give us a set of segmented images with different degrees of segmentation, from which to choose the one we want for our specific application, then we do not have to set a threshold value, but we do need to choose the segmentation we want. So in both cases there is a human intervention, at the start or at the end. That means that this is not a fully automatic segmentation method.

The fact that the user has the possibility of looking at various segmentation levels as a result of processing the image only once, makes the method versatile and gives it a dynamic characteristic, giving him the possibility of choosing the segmentation level that suits his particular needs. An important characteristic of this method is that it can work fairly well with images that show little contrast. This can be seen in Figure 6.

As future studies, the authors intend to investigate the efficiency of this method, in terms of computational resources.

Another aspect to study is obtaining optimal values for the optimal threshold parameter.

Also the authors are going to perform comparisons of the results of this method with other comparable methods.

References

1. Allende, H., Galbiati, J.: A non-parametric filter for digital image restoration, using cluster analysis. Pattern Recognition Letters 25, 841–847 (2004)
2. Allende, H., Galbiati, J.: Edge detection in contaminated images, using cluster analysis. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 945–953. Springer, Heidelberg (2005)
3. Allende, H., Becerra, C., Galbiati, J.: A segmentation method for digital images based on cluster analysis. Lecture Notes In Computer Science: Advanced And Natural Computing Algorithms, vol. 4431(1), pp. 554–563 (2007)
4. Frucci, M., Ramella, G., di Baja, G.S.: Using resolution pyramids for watershed image segmentation. Image and Vision Computing 25(6), 1021 (2007)
5. Fukui, M., Kato, N., Ikeda, H., Kashimura, H.: Size-independent image segmentation by hierarchical clustering and its application for face detection. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 686–693. Springer, Heidelberg (2004)

6. Martin, D., Fowlkes, C., Tal, D., Malikand, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring. In: Ecological Statistics, Proc. 8th Int'l Conf. on Computer Vision, vol. 2, pp. 416–423 (2001)
7. Martinez-Uso, A., Pla, F., Garcia-Sevilla, P.: Unsupervised image segmentation using a hierarchical clustering selection process. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 799–807. Springer, Heidelberg (2006)
8. Pan, Z., Lu, J.: A Bayes-based region-growing algorithm for medical image segmentation. Computing In Science and Engineering 9(4), 32–38 (2007)
9. Pauwels, J., Frederix, G.: Finding salient regions in images. Computer Vision and Image Understyanding 75, 73–85 (1999)
10. Wang, K.B., Yu, B.Z., Zhao, J., Li, H.N., Xie, H.M.: Texture image segmentation based on Gabor wavelet using active contours without edges. Department of Electronic Engineering, Northwestern Polytechnical University, Xi'an 710072, China

A Study on Representations for Face Recognition from Thermal Images

Yenisel Plasencia[1,2], Edel García-Reyes[1], Robert P.W. Duin[2],
Heydi Mendez-Vazquez[1], César San-Martin[3], and Claudio Soto[3]

[1] Advanced Technologies Application Center. 7a ♯ 21812 e/ 218 y 222, Rpto. Siboney,
Playa, C.P. 12200, La Habana, Cuba
{yplasencia,hmendez,egarcia}@cenatav.co.cu

[2] Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands
r.duin@ieee.org

[3] Information Proc. Lab, DIE, Universidad de La Frontera. Casilla 54-D,
Temuco, Chile
csmarti@ufro.cl, c.soto04@ufromail.cl

Abstract. Two challenges of face recognition at a distance are the uncontrolled illumination and the low resolution of the images. One approach to tackle the first limitation is to use longwave infrared face images since they are invariant to illumination changes. In this paper we study classification performances on 3 different representations: pixel-based, histogram, and dissimilarity representation based on histogram distances for face recognition from low resolution longwave infrared images. The experiments show that the optimal representation depends on the resolution of images and histogram bins. It was also observed that low resolution thermal images joined to a proper representation are sufficient to discriminate between subjects and we suggest that they can be promising for applications such as face tracking.

Keywords: face recognition, dissimilarity representations, thermal infrared.

1 Introduction

The use of longwave infrared (LWIR) imagery ($8 - 12\mu$m) for developing face recognition systems has started to receive attention in the last years because of its robustness to illumination changes [1]. LWIR (also called thermal infrared) sensors collect the heat energy emitted by a body instead of the reflected light, this allows them to operate even in complete darkness. The use of the face thermal patterns has been validated as a biometric signature on short time scale. We are interested in the study of such imagery modality for face recognition at a distance, what implies to handle low resolution images.

The opaqueness to glass is one of the limitations of thermal infrared face recognition. Studies in the literature show that there is a preference for fusing

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 185–192, 2009.

thermal and visible imagery to tackle this problem [2,3,4]. Another disadvantage of LWIR face representation is that it is sensitive to body temperature changes. Such changes can be provoked by external temperature like cold or warm air, by body exercising, or simply by consuming alcoholic beverages. However, for an application such as short term face tracking, the use of this imagery modality can be beneficial because temperature changes are negligible.

A number of algorithms has been proposed for the classification of faces from LWIR images, but in the representation level, only feature representations have been studied. Three major groups of methods can be distinguished for representation in the pattern recognition literature. The first group treats objects as vectors in Euclidean vector spaces. The second represents the object structure by graphs, grammars, etc. The third and newest group represents the objects by their nearness relation with respect to other objects. The vector space representation is the most developed one due to the existence of several statistical techniques that have shown a good performance on the different pattern recognition problems. Pekalska et al. [5] proposed the dissimilarity representation. It arises from the idea that the classes are constituted by similar objects, so (dis)similarities are more important than features for the class constitution. This dissimilarity approach has the potential of unifying the statistical and the structural approaches [6] because for example, statistical techniques can be used on a dissimilarity representation derived from graphs.

In this paper we will study classification performances on a dissimilarity space based on histogram distances, on the feature space of the histograms, and on the pixel space for face recognition from low resolution longwave infrared images. We will study in which conditions one representation is better than the other. Section 2 presents related work in face recognition using dissimilarity representations. Section 3 introduces the dissimilarity space representation. Histograms and the Chi Square distance are briefly described in Section 3. Section 4 presents the experimental results, including data, experimental setup, and discussion. The conclusions are drawn in Section 5.

2 Related Work

There are some studies in the literature where dissimilarity representations are introduced for face recognition, but none of them make use of thermal infrared imagery. In [7] after reducing dimensionality by Principal Component Analysis (PCA), the authors used the Euclidean distances to conform the dissimilarity matrix that characterizes the face data. They built a dissimilarity space from the Euclidean distances derived from a feature space. Then they compared linear and quadratic classifiers in that space with the nearest neighbor (1-NN) classifier applied directly to the dissimilarity matrix, as a function of the amount of prototypes selected per class. In their experiments they showed that the dissimilarity space classifiers outperformed the 1-NN rule.

In [8], the author proposed the use of dissimilarity representations to solve the Small Sample Size problem that affects the direct application of the Linear

Discriminant Analysis (LDA) method for face recognition. This is an alternative to the conventional use of methods like PCA as a previous step before the application of LDA. The method joined to a classifier fusion strategy was proved in face recognition and the results were comparable to the state of the art results.

3 Dissimilarity Space

The dissimilarity space was proposed by Pekalska et al. [5]. It was postulated as a Euclidean vector space. For its construction a representation set $R = \{r_1, r_2, ..., r_n\}$ is needed, where the objects belonging to this set (also called prototypes) are chosen adequately based on some criterion that can be dependent of the problem at hand. Let X be the training set, R and X can have the following relationships: $R \cap X = \varnothing$, or $R \subseteq X$. Once we have R, the dissimilarities of the objects in X to the objects in R are computed. When a new object r comes, it is also represented by a vector of dissimilarities d_r to the objects in R (1).

$$d_r = [d(r, r_1)d(r, r_2)...d(r, r_n)] \ . \tag{1}$$

The dissimilarity space is defined by the set R so each coordinate of a point in that space corresponds to a dissimilarity to some prototype and the dimension of this space is determined by the amount of prototypes selected. This allows us to control the computational cost and to guarantee the trade off between classification accuracy and computational efficiency.

3.1 Histograms

Before computing the dissimilarity values for the creation of the dissimilarity space, pixel intensity histograms of the whole image were used as an intermediate representation. In our approach, the use of histograms has the advantage of allowing horizontal shifts and rotations of the face in the image. As it is shown in Fig. 1, in the face images selected for our experiments the background is almost constant with some exceptions like the nonuniformity noise. Also the majority of the background pixel intensities are different from the face pixel intensities, implying that the background information is not supposed to interfere with the face information.

3.2 Chi Square Distance

For the comparison of the LWIR histograms, the Chi Square distance measure [9] was used. This distance has been proving to be effective for histogram comparison. Let S and M be two histograms and n the number of bins in the histogram. The Chi square distance is defined as follows:

$$\chi^2(S, M) = \sum_{i=1}^{n} \frac{(S_i - M_i)^2}{(S_i + M_i)} \ . \tag{2}$$

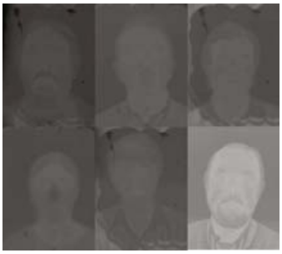


Fig. 1. Examples of LWIR images from the Equinox database

4 Experiments

Our goal is to compare a pixel-based, a histogram representation and a dissimilarity representation based on the histograms by means of classification accuracies of 1-NN and LDA classifiers for thermal infrared face recognition in the presence of low resolution images.

4.1 Data and Experimental Setup

For the experiments the Equinox face database was used, which is a benchmark database for thermal infrared face recognition. It was collected by Equinox Corporation under DARPAs HumanID program [10]. In total the database contains grayscale images of 89 subjects with 12 bits per pixel and 320x240 pixels of size.

The methodology described in [11] was followed for the experiments, but the subsets containing the pictures of the subjects wearing glasses were discarded. Different subsets were considered using (F)rontal and (L)ateral illuminations. VF and VL are (V)owel subsets including images of the subjects moving the mouth to pronounce the vowels. EF and EL are (E)xpression subsets including images of the subjects with different expressions. VA is composed of images taken from VF and VL. EA is composed of images taken from EF and EL. The experimental setup is shown in Table 1. Each subset used for training and test contains 267 images (3 images per 89 subjects).

In LWIR face images there is a lack of accurate techniques for detecting face landmark points. These points are needed for the geometric normalization of the face. We try to overcome this limitation using an histogram based representation that is robust to head rotations and horizontal shifts of the face in the scene. For the experiments 5 different image sizes were considered: 320x240, 80x60, 32x24, 16x12, and 6x8 pixels. An example of images with the 320x240 and 16x12 resolution and their related histograms is shown in Fig. 2 and Fig. 3.

As a reference we tested the pixel-based representation without a geometric normalization of the face. For the histogram representation the number of bins is 256. Histograms were normalized with respect to the number of pixels of the image. The dissimilarity representation was conformed using the Chi Square distance between the histograms. As representation set for projecting the patterns in the dissimilarity space we used the entire training set.

Table 1. Experimental setup

Setup	Training set	Test sets	Result
VL/VF	VL	VF1, VF2, VF3	a_1 = Average(VL/VF1,VL/VF2,VL/VF2)
VF/VA	VF	VA1, VA2, VA3	a_2 = Average(VF/VA1,VF/VA2,VF/VA2)
VL/VA	VL	VA1, VA2, VA3	a_3 = Average(VL/VA1,VL/VA2,VL/VA2)
VF/VL	VF	VL1, VL2, VL3	a_4 = Average(VF/VL1,VF/VL2,VF/VL2)
VF/EF	VF	EF1, EF2	a_5 = Average(VF/EF1,VF/EF2)
VA/EF	VA	EF1, EF2	a_6 = Average(VA/EF1,VA/EF2)
VL/EF	VL	EF1, EF2	a_7 = Average(VL/EF1,VL/EF2)
VF/EA	VF	EA1, EA2	a_8 = Average(VF/EA1,VF/EA2)
VA/EA	VA	EA1, EA2	a_9 = Average(VA/EA1,VA/EA2)
VL/EA	VL	EA1, EA2	a_{10} = Average(VL/EA1,VL/EA2)
VF/EL	VF	EL1, EL2	a_{11} = Average(VF/EL1,VF/EL2)
VA/EL	VA	EL1, EL2	a_{12} = Average(VA/EL1,VA/EL2)
VL/EL	VL	EL1, EL2	a_{13} = Average(VL/EL1,VL/EL2)

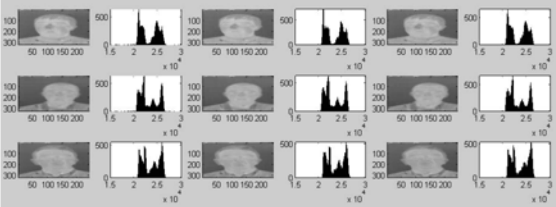


Fig. 2. Examples of images with resolution of 320x240 and their related histograms. Each row contains images of one subject.

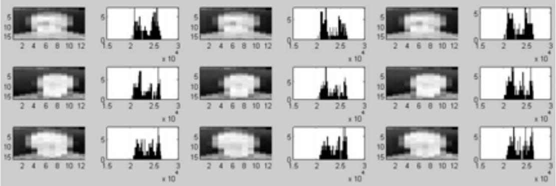


Fig. 3. Examples of images with resolution of 16x12 and their related histograms. Each row contains images of one subject.

Table 2 shows the results in terms of means and standard deviations of classification accuracies of 1-NN and LDA classifiers for the 3 different representations and the 5 different image resolutions. This classifiers are very different since LDA is linear and global and 1-NN is highly nonlinear and a local classifier. Each mean is calculated over the 13 results $(a_1, a_2, ..., a_{13})$ in Table 1. In all the experiments the data was reduced to 48 dimensions with PCA in order to avoid regularization parameters for LDA and make the approaches comparable. The 1-NN classifier uses the Euclidean distance for all representations. The results where the accuracy is statistically significantly higher for each classifier and resolution for the different representations are in bold. This is evaluated using the formula $\gamma = |\mu_1 - \mu_2| - \frac{(\sigma_1+\sigma_2)}{\sqrt{N}}$, where μ_1 and μ_2 are the two means of classifier

Table 2. Average classification accuracies and (standard deviations) for 1-NN and LDA classifiers on a dissimilarity, a histogram, and a pixel-based representation for different image resolutions

Resolution	Classifier	Dissimilarity Rep	Histogram Rep	Pixel-based Rep
320x240	LDA	91.88 (10.78)	86.01 (15.91)	78.20 (21.62)
	1-NN	99.34 (0.85)	**99.73 (0.25)**	95.17 (2.31)
80x60	LDA	95.24 (6.66)	91.52 (10.98)	62.61 (26.14)
	1-NN	99.64 (0.41)	99.67 (0.34)	95.24 (2.30)
32x24	LDA	**99.53 (0.78)**	97.14 (3.41)	66.81 (25.72)
	1-NN	**99.57 (0.28)**	99.27 (0.48)	95.40 (2.29)
16x12	LDA	98.07 (1.22)	98.38 (1.47)	68.95 (27.04)
	1-NN	**97.34 (1.09)**	94.26 (2.47)	95.83 (2.12)
8x6	LDA	75.53 (7.82)	76.21 (7.82)	71.79 (25.67)
	1-NN	66.73 (7.65)	53.64 (8.53)	**96.32 (2.02)**

accuracies, and σ_1 and σ_2 are the standard deviations and N is the number of experiments. Taking a threshold $\alpha = 0.5$ the difference on the means is statistically significant if $\gamma > \alpha$.

4.2 Results and Discussion

Our classification results using the 1-NN on the 320x240 images, using both the histogram and the dissimilarity representations, are comparable to or better than previous results reported in the literature. For example in [1] the authors tested the LDA method on the geometrically normalized 324x240 images and the average classification accuracy was 96.78. In our case, despite the fact that no geometric normalization was made, with the 1-NN on the dissimilarity space and the 1-NN on the histograms we achieved 99.34 and 99.73 of accuracy on the same data.

For the 320x240 resolution we can see that the two classifiers on the pixel-based representation suffer from the fact that the faces are not aligned, so classification results are the lowest. By using histograms, we achieve invariance to horizontal shifts and rotations of the face inside the image, leading to better classification results than with the pixel-based representation.

For 80x60, 32x24, and 16x12 resolutions, classification results on dissimilarity and histogram representations continue to be better than classification results on the pixel-based representations. Also for this low resolutions, classification accuracies are statistically significantly higher when using dissimilarity representations and in 2 of 3 cases the highest accuracies correspond to the 1-NN on the dissimilarity representation. By decreasing the resolution, histograms are more sparse so bins become less reliable features. Dissimilarities can handle this a little bit better. Also, the LDA classifier performs better because is a more global classifier and suffer less from this noise.

For the 32x24 resolution we can observe that classification results on both the histogram and the dissimilarity representations are similar to or better than classification results on the high resolution images. This may suggest that for recognizing faces from LWIR images we do not need a high resolution for the

Table 3. Average classification accuracies (and standard deviations) when decreasing the resolution of histograms to 50, 20, 10, and 5 bins for the 8x6 image resolution

Classifier	Dissimilarity Rep	Histogram Rep
LDA hist 50 bins	93.13 (3.29)	91.18 (4.40)
1-NN hist 50 bins	86.94 (4.36)	86.28 (4.39)
LDA hist 20 bins	87.50 (4.38)	92.97 (2.76)
1-NN hist 20 bins	86.01 (4.28)	84.23 (4.68)
LDA hist 10 bins	79.68 (5.13)	86.17 (3.92)
1-NN hist 10 bins	77.71 (6.16)	77.11 (6.12)
LDA hist 5 bins	30.73 (27.65)	64.56 (4.46)
1-NN hist 5 bins	55.22 (5.82)	58.30 (6.20)

images. This conclusion can only be applicable to databases with a controlled size like in our setups.

In the case of 8x6 images, classifiers on histogram and dissimilarity representations perform very poor compared to the 1-NN on the pixel-based representation. The pixel-based representation is performing better because by decreasing the resolution, the faces become aligned. On the other hand, the poor performance of histograms and dissimilarities can be attributed to the fixed number of histogram bins that leads to very sparse histograms when the resolution is as low as 8x6. To prove this, we conducted some experiments diminishing the number of bins of the histograms. The new bin values are the summations over neighboring bins of the larger histograms. The resolution of the histograms was decreased to 50, 20, 10, and 5. The results are shown in Table 3. We can observe that classification results can be improved if the histogram resolution is decreased while the image resolution is decreased (e.g. 50, 20, and 10 bins), but we need to take care of the selected histogram size because classification accuracy starts to decrease for very small sizes. When using histograms of 5 bins classification results are no longer better than those using 256 bins. The best result is obtained with LDA on the dissimilarity representation using histograms of 50 bins.

5 Conclusions

It is very common to find studies on complicated representations for face recognition from visual imagery. This is needed in order to achieve invariance to factors that affect visual images and degrade the recognition performance such as ambient illumination changes. By using the thermal imagery modality, invariance to illumination changes is achieved. Therefore, very simple representations can be suitable for this type of images.

In this paper we compared some representations such as a pixel-based, a histogram representation and a dissimilarity representation based on the histograms for face recognition from low resolution LWIR images. We find out that histograms characterize the subjects sufficiently, and dissimilarities on the histograms may improve this. For low resolutions images, histograms become sparse and results deteriorate. The pixel-based representation can now perform very well because the faces become more aligned. A good tradeoff between image resolution and histogram resolution may be needed. It was also observed that low resolution thermal images joined to a proper representation are sufficient to discriminate between subjects and we suggest that they can be used in practical applications such as face tracking.

Acknowledgements. We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Socolinsky, D.A., Selinger, A.: A comparative analysis of face recognition performance with visible and thermal infrared imagery. In: ICPR 2002, Washington, DC, USA, vol. 4, p. 40217. IEEE Computer Society, Los Alamitos (2002)
2. Bebis, G., Gyaourova, A., Singh, S., Pavlidis, I.: Face recognition by fusing thermal infrared and visible imagery. Image Vision Comput. 24(7), 727–742 (2006)
3. Heo, J.: Fusion of visual and thermal face recognition techniques: A comparative study. University of Tennessee (2005)
4. Kong, S.G., Heo, J., Abidi, B.R., Paik, J., Abidi, M.A.: Recent advances in visual and infrared face recognition - a review. Computer Vision and Image Understanding 97, 103–135 (2005)
5. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc., River Edge (2005)
6. Bunke, H., Riesen, K.: Graph classication based on dissimilarity space embedding. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 996–1007. Springer, Heidelberg (2008)
7. Orozco-Alzate, M., Castellanos-Domínguez, C.G.: Nearest feature rules and dissimilarity representations for face recognition problems. In: Kurihara, K. (ed.) Face Recognition; International Journal of Advanced Robotic Systems, Vienna, Austria 337–356 (May 2007)
8. Kim, S.W.: On using a dissimilarity representation method to solve the small sample size problem for face recognition. In: ACIVS, pp. 1174–1185 (2006)
9. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: ECCV, vol. (1), pp. 469–481 (2004)
10. `http://www.equinoxsensors.com/products/HID.html`
11. Socolinsky, D.A., Wolff, L.B., Neuheisel, J.D., Eveland, C.K.: Illumination invariant face recognition using thermal infrared imagery. In: CVPR, vol. (1), pp. 527–534 (2001)

Fast Unsupervised Texture Segmentation Using Active Contours Model Driven by Bhattacharyya Gradient Flow

Foued Derraz[1,2], Abdelmalik Taleb-Ahmed[1], Antonio Pinti[1], Laurent Peyrodie[4], Nacim Betrouni[3], Azzeddine Chikh[2], and Fethi Bereksi-Reguig[2]

[1] LAMIH UMR CNRS 8530, Le Mont Houy, 59313 Valenciennes, France
[2] GBM Laboratory Abou Bekr Belkaid university Tlemcen, 13000, Algeria
[3] LAMIH UMR CNRS 8530, Le Mont Houy, 59313 Valenciennes
[4] Hautes Etudes d'Ingénieur Lille, France
f_derraz@hotmail.com, taleb@univ-valenciennes.fr,
Antonio.pinti@univ-valenciennes.fr, laurent.peyrodie@hei.fr,
n_betrouni@chu-lille.fr, az_chikh@hotmail.com,
f_bereksi@mail.univ-tlemcen.dz

Abstract. We present a new unsupervised segmentation based active contours model and texture descriptor. The proposed texture descriptor intrinsically describes the geometry of textural regions using the shape operator defined in Beltrami framework. We use Bhattacharyya distance to discriminate textures by maximizing distance between the probability density functions which leads to distinguish textural objects of interest and background. We propose a fast Bregman split implementation of our segmentation algorithm based on the dual formulation of the Total Variation norm. Finally, we show results on some challenging images to illustrate segmentations that are possible.

Keywords: Active contours, texture descriptor, bhattacharyya distance, total variation, Bregman split algorithm.

1 Introduction

Active contour models such as Geometric/Geodesic Active Contour (GAC) [1], Active Contours Without Edge (ACWE) [7] model have been widely used as image segmentation methods and more recently for texture segmentation [4]. Later, extension and generalization [8],[13],[16],[18]-[22], respectively, for vector-valued images has been done by replacing the scalar gray-level intensities with vectors of color channel intensities to guide contour evolution. However, the information derived from intensity integral operations deceived the textural image segmentation process as regions of different textures may have equal average intensities. Therefore the application ACWE model based on image intensities either in its original or in its generalized form can be considered unsuitable for texture segmentation. However, its region-based formulation could be exploited for capturing textural information derived from features not necessarily exhibiting high gradients at object boundaries.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 193–200, 2009.

In this paper, we proposed fast unsupervised segmentation algorithm to textural segmentation model based on Bhattacharyya distance.

The rest of paper is organized as follows. Firstly, we introduce the texture region and shape descriptor. Then, we define the active contour model based on the Bhattacharyya distance. We prove the existence of a minimizer. We then present the fast algorithm to determine evolving contour curve. Finally, we show some promising experimental results.

2 Texture Descriptor

In Beltrami framework [2], a smoothed version of an original gray level image $I:\mathbb{R}^2 \rightarrow \mathbb{R}^+$ can be viewed as a surface Σ with local coordinates (x,y) embedded in $\mathbb{R}^3$ by smooth mapping X. Let $(x,y) \rightarrow X = (x,y,G_\sigma * I)$, G_σ is Gaussian filter with σ^2 variance, the first fundamental form is define by:

$$g_{\mu,\nu}(x,y) = \left(\left\langle \frac{\partial X}{\partial \mu}, \frac{\partial X}{\partial \nu} \right\rangle \right) = \begin{pmatrix} 1+\hat{I}_x^2 & \hat{I}_x \hat{I}_y \\ \hat{I}_x \hat{I}_y & 1+\hat{I}_y^2 \end{pmatrix} \tag{1}$$

where $\mu,\nu = x,y$ in the (x,y)-basis and $\hat{I}_x$ and $\hat{I}_y$ in (1) are the image derivatives convolved with a relatively large Gaussian filter, such $\hat{I}_x = G_\sigma * I_x$ and $\hat{I}_y = G_\sigma * I_y$.

The inverse determinant (det) of metric tensor $g_{\mu,\nu}$ is defined as:

$$g(I) = \frac{1}{\det\left(g_{\mu,\nu}(x,y)\right)} = \frac{1}{1+\left|\nabla G_\sigma * I\right|^2} \tag{2}$$

Instead of this edge descriptor [3], we propose to define a region descriptor for textural image. For this, we design an intrinsic descriptor based on the use of shape operator to describe the geometry of textures [8]. The shape operator S measures the shape of the manifold in a given region by estimating how the normal N_Σ to the surface Σ changes from point to point. For a tangent vector v_p to Σ at p, the shape operator that satisfying [8]:

$$S = -D_{v_p} N_\Sigma \tag{3}$$

where $D_{v_p} N_\Sigma$ is the derivative of the surface normal in the direction v_p.

The second fundamental form $b_{\mu,\nu}$, used to measure oriented distance on manifolds, and its components indicate the direction of change of the manifold as:

$$b_{\mu,\nu}(x,y)=\left(\left\langle N_{\Sigma},\frac{\partial^2 X}{\partial\mu\partial\nu}\right\rangle\right)=\frac{1}{\sqrt{1+\hat{I}_x^2+\hat{I}_y^2}}\begin{pmatrix}\hat{I}_{xx} & \hat{I}_{xy}\\ \hat{I}_{xy} & \hat{I}_{yy}\end{pmatrix} \qquad (4)$$

where $N_{\Sigma}=\frac{1}{\sqrt{1+\hat{I}_x^2+\hat{I}_y^2}}\left(-\hat{I}_x,-\hat{I}_y,1\right)$ is normal to surface manifold.

The shape operator S calculates the bending of a smoothed surface in different directions [8]. The principles curvatures of the manifold are the roots of the following equation:

$$k^2-b_{\mu\nu}(x,y)\,g^{\mu\nu}(x,y)\,k+\frac{b}{g}=0 \qquad (5)$$

where $g^{\mu\nu}(x,y)=g_{\mu\nu}{}^{-1}(x,y)$, $b=\det\left(b_{\mu,\nu}(x,y)\right)$ and $g=\det\left(g_{\mu,\nu}(x,y)\right)$.

The first principal curvature $\kappa_{\max}$ corresponds to the maximal change of the normal to the surface and $\kappa_{\min}$ corresponds to the minimum change:

$$\kappa_{\max}=-\frac{1}{2}trace\left(b_{\mu,\nu}(x,y)\,g^{\mu,\nu}(x,y)\right)+\sqrt{\frac{1}{4}\left(trace\left(b_{\mu,\nu}(x,y)\,g^{\mu,\nu}(x,y)\right)\right)^2-\frac{b}{g}} \qquad (6)$$

$$\kappa_{\min}=-\frac{1}{2}trace\left(b_{\mu,\nu}(x,y)\,g^{\mu,\nu}(x,y)\right)-\sqrt{\frac{1}{4}\left(trace\left(b_{\mu,\nu}(x,y)\,g^{\mu,\nu}(x,y)\right)\right)^2-\frac{b}{g}} \qquad (7)$$

where $b=\frac{1}{Z}\left(\hat{I}_{xx}\hat{I}_{yy}-\hat{I}_{xy}^2\right)$, $g=Z^2$, $Z=\sqrt{1+\hat{I}_x^2+\hat{I}_y^2}$

and $b_{\mu,\nu}(x,y)\,g^{\mu,\nu}(x,y)=\frac{1}{gZ}\begin{pmatrix}\hat{I}_{xx}\left(1+\hat{I}_y^2\right)+\hat{I}_{xy}\hat{I}_x\hat{I}_y & \hat{I}_{xy}\left(1+\hat{I}_x^2\right)+\hat{I}_{xx}\hat{I}_x\hat{I}_y\\ \hat{I}_{xy}\left(1+\hat{I}_y^2\right)+\hat{I}_{yy}\hat{I}_x\hat{I}_y & \hat{I}_{yy}\left(1+\hat{I}_x^2\right)+\hat{I}_{xy}\hat{I}_x\hat{I}_y\end{pmatrix}$,

Since $\kappa_{\max}\perp\kappa_{\min}$, we propose to define the texture descriptor as:

$$\kappa_T=\operatorname{atan}\left(\frac{\kappa_{\max}}{\kappa_{\min}}\right) \qquad (8)$$

Where $\kappa_T:\Omega\rightarrow\mathbb{R}^+$ is used to segment regions with different texture patterns, Ω corresponds to the image domain.

3 Bhattacharyya Flow

The Bhattacharyya distance between two probability density functions is defined as $E_{Bat}=-\log\left(Bat\right)$, where Bat is the Bhattacharyya coefficient given by [16,17]:

$$Bat\left(p_{in},p_{out}\right)=\int_{R^+}\sqrt{p_{in}\left(\kappa_T,\Omega\right)p_{out}\left(\kappa_T,\Omega\right)}d\kappa_T \tag{9}$$

The pdfs p_{in} and p_{out} associated with an observation κ_T for a fixed region Ω are defined by the Parzen kernel:

$$\begin{cases} p_{in}\left(\kappa_T,\Omega\right)=\dfrac{1}{|\Omega|}\displaystyle\int_{\Omega}K_{\sigma_{\text{ker}}}\left(\kappa_T-\kappa_T\left(x\right)\right)dx \\ p_{out}\left(\kappa_T,\Omega\right)=\dfrac{1}{|\Omega_0\setminus\Omega|}\displaystyle\int_{\Omega_0\setminus\Omega}K_{\sigma_{\text{ker}}}\left(\kappa_T-\kappa_T\left(x\right)\right)dx \end{cases} \tag{10}$$

In order to produce two regions, the object Ω and the background $\Omega_0\setminus\Omega$, with two pdfs as disjoint as possible, the energy the functional E_{Bat} is maximized, w.r.t the evolving domain $\Omega\left(t\right)$, is done with the shape derivative tool[4,10]. Thus, the Eulerian derivative of E_{Bat} in the direction ξ is as follows:

$$\left\langle\frac{\partial E_{Bat}\left(\Omega(t)\right)}{\partial t},\xi\right\rangle=\int_{\partial\Omega}V_{Bat}\left\langle\xi\left(s\right),N_C\left(s\right)\right\rangle ds \tag{11}$$

where the Bhattacharyya velocity is expressed as:

$$\begin{aligned} V_{Bat}=&\frac{1}{2}\left(\frac{1}{|\Omega|}-\frac{1}{|\Omega_0\setminus\Omega|}\right)\sqrt{P_{in}\left(\kappa_T,\Omega\right)P_{out}\left(\kappa_T,\Omega\right)} \\ &+\frac{1}{2|\Omega_0\setminus\Omega|}\int_{R^+}\sqrt{\frac{P_{in}\left(\kappa_T,\Omega\right)}{P_{out}\left(\kappa_T,\Omega\right)}}\left(\begin{matrix}K_{\sigma_{\text{ker}}}\left(\kappa_T-\kappa(s)\right)\\ -\sqrt{P_{out}\left(\kappa_T,\Omega\right)}\end{matrix}\right)d\kappa_T \\ &-\frac{1}{2|\Omega|}\int_{R^+}\sqrt{\frac{P_{out}\left(\kappa_T,\Omega\right)}{P_{in}\left(\kappa_T,\Omega\right)}}\left(\begin{matrix}K_{\sigma_{\text{ker}}}\left(\kappa_T-\kappa(s)\right)\\ -\sqrt{P_{in}\left(\kappa_T,\Omega\right)}\end{matrix}\right)d\kappa_T \end{aligned} \tag{12}$$

Where $\vec{N}$ is an exterior unit normal vector to the boundary $C=\partial\Omega$ of the region Ω, $\left\langle\varepsilon,\vec{N}\right\rangle$ is the Euclidean scalar product and s is the arc length parametrization. If we consider the energy functional expressed as:

$$E\left(\Omega\right)=L_g\left(\Omega\right)+\lambda E_{Bat}\left(\Omega\right) \tag{13}$$

where $L_g\left(\Omega\right)$ is the length of the boundary of Ω and acts like a regularization process in the curve evolution, λ positive constant which controls the trade-off between the regularization process and the fidelity of the solution. In Total variation norm the energy of active contours can be expressed as:

$$\min_{u\in[0,1]}\left(E(u)\right)=\lambda\int_{\Omega_0} V_{Bat}u+\int_{\Omega_0} g(I)\left|\nabla u\right| \tag{14}$$

In the next section we introduced a fast algorithm for solving segmentation problem.

4 Fast Algorithm Based on Split Bregman

A fast and accurate minimization algorithm for (16) is introduced in [5]. We substitute ϕ by u to formulate the variational problem:

$$\min_{u\in[0,1]}\left(E(u)\right)=\lambda\int_{\Omega_0} V_{Bat}u+\int_{\Omega_0} g(I)\left|\nabla u\right| \tag{15}$$

A new vectorial function d is introduced as follows:

$$\min_{u\in[0,1],d}\left(\lambda\int_{\Omega_0} V_{Bat}u+\int_{\Omega_0} g(I)\left|d\right|\right) \tag{16}$$

The constraintis $d=\nabla u$ enforced using the efficient Bregman iteration approach [10, 13, 3] defined as:

$$\begin{cases}\left(u^{k+1},d^{k+1}\right)=\arg\min\left(\left\{\lambda\int_{\Omega_0} V_{Bat}u+\int_{\Omega_0} g(I)\left|d\right|+\frac{\mu}{2}\int_{\Omega_0}\left|d-\nabla u-b^k\right|^2\right\}\right)\\ \qquad b^{k+1}=b^k+\nabla u^k-d^{k+1}\end{cases} \tag{17}$$

The minimizing solution u^{k+1} is characterized by the optimality condition:

$$\mu\Delta u=\lambda V_{bat}+\mu div\left(b^k-d^k\right),u\in[0,1] \tag{18}$$

A fast approximated solution is provided by a Gauss-Seidel iterative scheme:

$$\begin{cases}\gamma_{i,j}=d^{x,k}_{i-1,j}-d^{x,k}_{i,j}-b^{x,k}_{i-1,j}+b^{x,k}_{i,j}+d^{y,k}_{i,j-1}-d^{y,k}_{i,j}-b^{y,k}_{i,j-1}+b^{y,k}_{i,j}\\ \mu_{i,j}=\frac{1}{4}\left(u^{k,n}_{i-1,j}+u^{k,n}_{i+1,j}+u^{k,n}_{i,j+1}+u^{k,n}_{i,j+1}-\frac{\lambda}{\mu}V_{Bat,i,j}+\gamma_{i,j}\right)\quad,n>0,k>0\\ \qquad u^{k+1,n+1}_{i,j}=\max\left\{\min\left\{\mu_{i,j},1\right\},0\right\}\end{cases} \tag{19}$$

Finally, the minimizing solution d^{k+1} is given by soft-thresholding:

$$d_{i,j}^{k+1} = \frac{\nabla u^{k+1} + b^k}{\left|\nabla u^{k+1} + b^k\right|} \max\left(\left|\nabla u^{k+1} + b^k\right| - \mu^{-1}, 0\right) \tag{20}$$

Then, the final active contour is given by the boundary of the set $\left\{\mathbf{x} \in \Omega \middle| u^{final} > \frac{1}{2}\right\}$. The two iteration schemes are straightforward to implement. Finally, we update at each iteration p_{in} , p_{out} using the Parzen kernel given in equation (10).

5 Experiments Results

We applied the proposed segmentation algorithm to a set of challenging real-world textural images (image a),b),c))). The natural textural images were taken in the Berkeley segmentation data set 14. Fig. 1 presents the results obtained with the proposed method. We notice that our segmentation model needs four parameters, σ explained in section 2. θ, λ explained in Section 4. The Parzen parameter in Section 3. The mean computing time for the segmentation is around a minute. The segmentation results are compared to manual segmentation [14], and we evaluation the quality of segmentation in term of F-measure detailed in [14]. For a good choice of the segmentation parameters, the results are compared to manual segmentation and the F-measure drawn (Table 1)an improvement of segmentation quality compared to results drawn by the model proposed in [3].

Table 1. Quatitative evaluation of the segmentation

Image	P	R	F
Image a	0.61	0.59	0.60
Image b	0.63	0.60	0.61
Image c	0.65	0.62	0.63

Integrating the texture region descriptor guide the active contour to localize efficiently the geometry of textured region. Solving the segmentation problem in dual TV allows the active contour to reach the minimum global and ensure the active contour to segment the one textural region in image. We have compared the segmentation results of image a), b), c) to the manual segmentation. The quality of segmentation expressed in F-measure term shows that the proposed method segments successfully the textured regions. An adequate choice of parameters model leads to a good segmentation of textural image.

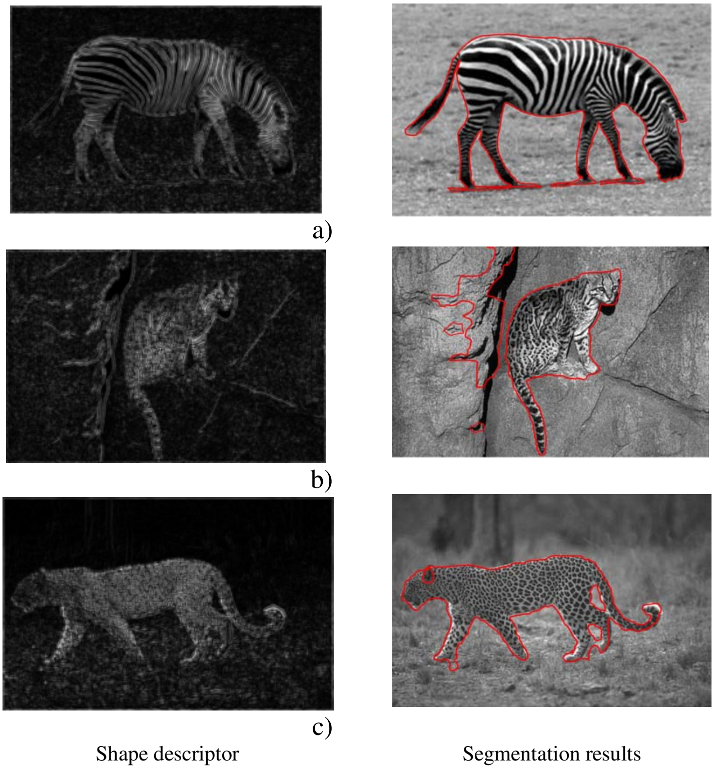


Fig. 1. Segmentation of textural images based on bhattacharyya distance

6 Conclusion

We have introduced an active contour model based Bhattacharyya gradient flow for unsupervised segmentation of textural images. We have proposed a new intrinsic textural feature descriptor based on the shape operator of the texture manifold and fast algorithm is developed based on a dual TV approach. The proposed model is designed to work with textures and needs at least one textural region.

References

1. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic Active Contours. IJCV 22(1), 61–97 (1997)
2. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE TIP 7(3), 310–318 (1996)
3. Sagiv, C., Sochen, N., Zeevi, Y.: Integrated active contours for texture segmentation. IEEE TIP 15(6), 1633–1646 (2006)
4. Aujol, J.-F., Gilboa, G., Chan, T., Osher, S.: Structure-Texture Image Decomposition-Modeling, Algorithms, and Parameter Selection. IJCV 67(1), 111–136 (2006)

5. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast Global Minimization of the Active Contour/Snake Model. JMIV 28(2), 51–167 (2007)
6. Mi, A.S., Iakovidis, D.K., Maroulis, D.: LBP-guided active contours. Pattern Recognition Letters 29(9), 1404–1415 (2008)
7. Chan, T., Vese, L.: Active Contours Without Edges. IEEE TIP 10(2), 266–277 (2001)
8. Delfour, M., Zolésio, J.: Shapes and Geometries: Analysis, Differential Calculus, and Optimization. Advances in Design and Control, SIAM (2001)
9. Freedman, D., Zhang, T.: Active contours for tracking distributions. IEEE TIP 13(4), 518–526 (2004)
10. Herbulot, A., Jehan-Besson, S., Duffiner, S., Barlaud, M., Aubert, G.: Segmentation of vectorial image features using shape gradients and information measures. JMIV 25(3), 365–386 (2006)
11. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. In: Proc. IEEE CVPR 2003, Madison, WI, USA, vol. 2, pp. 699–704 (2003)
12. Lee, S.M., Abott, A.L., Clark, N.A., Araman, P.A.: Active contours on statistical manifolds and texture segmentation. In: Proc. IEEE ICIP 2005, vol. 3, pp. 828–831 (2005)
13. Chan, T., Sandberg, B., Vese, L.: Active contours without edges for vector-valued images. JVCIR 11(2), 130–141 (2000)
14. Martin, D., Fowlkes, C., Malik, J.: Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. IEEE PAMI 26(5), 530–549 (2004)
15. Goldstein, T., Bresson, X., Osher, S.: Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction. In: Technical Report 06, Math. Department UCLA, Los Angeles, USA (2009)
16. Michailovich, O., Rathi, Y., Tannenbaum, A.: Image Segmentation Using Active Contours Driven by the Bhattacharyya Gradient Flow. IEEE TIP 16(11), 2787–2801 (2007)
17. Raubera, T.W., Braunb, K.B.: Probabilistic distance measures of the Dirichlet and Beta distributions. Pattern Recognition 41(2), 637–645 (2008)
18. Lecellier, F., Fadili, J., Jehan-Besson, S., Aubert, G., Revenu, M.: Region-based active contours and sparse representations for texture segmentation. In: Proc. IEEE ICPR 2008, Florida (2008)
19. Allili, M.S., Ziou, D., Bentabet, L.: A robust level set approach for image segmentation and statistical modeling. In: Proc. Adv. Conc. on Intelligent Vision Systems (ACIVS), pp. 243–251 (2004)
20. Sandberg, B., Chan, T., Vese, L.: A level-set and gabor-based active contour algorithm for segmenting textured images. In: Technical Report 39, Math. Department UCLA, Los Angeles, USA (2002)
21. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algoruithms for l1 minimization with applications to compressed sensing. SIAM J. Imaging Sci. 1, 143–168 (2008)
22. He, Y., Luo, Y., Hu, D.: Unsupervised Texture Segmentation via Applying Geodesic Active Regions to Gaborian Feature Space. IEEE Transactions on Engineering, Computing and Technology, 272–275 (2004)

A Distributed and Collective Approach for Curved Object-Based Range Image Segmentation

Smaine Mazouzi[1], Zahia Guessoum[2], and Fabien Michel[3]

[1] Dép. d'informatique, Université de Skikda, BP 26, Route ElHadaik, 21000, Algérie
mazouzi_smaine@yahoo.fr
[2] LIP6, Université de Paris 6, 104, av. du Président Kennedy, 75016, Paris, France
zahia.guessoum@lip6.fr
[3] LIRMM 161 rue Ada 34392 Montpellier Cedex 5, France
fmichel@lirmm.fr

Abstract. In this paper, we use multi-agent paradigm in order to propose a new method of image segmentation. The images considered in this work are the range images which can contain at once polyhedral and curved objects. The proposed method uses a multi-agent approach where agents align the region borders to the surrounding surfaces which make emerging a collective segmentation of the image. The agents move on the image and when they arrive on the pixels of a region border they align these pixels to their respective surfaces. The resulting competitive alignment allows at once the emergence of the image edges and the disappearance of the noise regions. The test results obtained with real images show a good potential of the new method for accurate image segmentation.

Keywords: Image segmentation, Multi-agent systems, Curved Object, Range image.

1 Introduction

A range image represents the visible surface of a three-dimensional scene, where at each pixel of the image is stored the depth of the corresponding point of the scene. Range images are mainly used in recognition of 3D objects in robotic vision, because the 3D information, which is required for object recognition is immediately available. However, these images are recognized as being highly noised images [3] and are consequently hard to segment and to interpret. Several authors proposed methods of segmentation for this class of images [4,8]. However, most of these authors limited themselves to the images based on polyhedral objects. For these last ones, the detection of the regions of interest is widely easier compared with images containing curved objects [4].

Aiming at segmenting range images with both polyhedral and curved objects, we have appealed to multi-agent paradigm and more particularly to the reactive agents [2]. We have used simple agents with weak granularity and having simple

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 201–208, 2009.

behavior which based on the mechanism of stimulus-reaction. The agents move on the image and act on the met pixels. An agent aligns the first not homogeneous pixel to the surface on which it moves. So, pixels belonging to the noise regions or situated on the circumferences of surfaces are systematically aligned to the surfaces which surround them. We show that the alternative alignment of pixels on the borders of regions allows preserving edges. On the other hand, because there is no pixel alignment within the noise regions, theses regions progressively disappear. This collective action of the agents allows the emergence of edges and the disappearance of the noise regions.

Most of the methods having used the agent paradigm [1,7,9,10,6] are supervised and consequently they can be applied only to the images for which they were conceived. The method proposed in this paper is unsupervised. It makes no assumption on the number and the shape of the surfaces which compose objects in the image. This allows the adaptation of the method to any type of images by defining the criteria of homogeneity of the regions composing the treated images. The distribution of treatments and decision which characterize our approach, as well as the weak coupling of the reactive agents, offers a parallel method well suitable for real-time image interpretation. The experimentation of the method by using real images allowed to validate the new approach and to show its potential for an accurate and effective segmentation of range images.

The reminder of the paper is organized as follows: in Section 2 we present the main approaches having used multi-agent systems for image segmentation. Section 3 is devoted to the proposed approach. We present at first the criterion of homogeneity curved surfaces adopted in our case. Then, we introduce the principle of the multi-agent system and we show how a collective segmentation emerges form the simple and reactive behaviors of the agents. Our test results are introduced in the section 5, in which we show the the parameter selection and we comment on the obtained results. Finally, a conclusion summarizes our work and underlines its potential extensions.

2 Multi-agent Approches for Image Segmentation

Since the publication of the first works proposing multi-agent approaches for image processing, the cooperation and the interaction between agents represented the new contributions to deal with the problem of ambiguity which characterizes image visual data [1,7].

In the category of edge based segmentation methods using a reactive approach, Ballet et al. [1] have defined a multi-agent system for edge detection and following in 2D images. In this system an agent is situated on an extremum of the luminance gradient, then follows the crest of the edge and records its path in a shared memory. The previously recorded edge segments are used by other agents to initialize new edge segments or finish others. The method proposed by the authors is based on an original approach of parallel and distributed edge detection. However, the authors have not considered any mechanism of cooperation or coordination to strengthen the detection and improve the segmentation results.

The same authors [10] have resumed later the same approach and have extended it by using the multi-agent language oRis. By considering the a priori knowledge on the topology of regions in certain types of images, the agents follow the lines of gradient extremum. By considering two types of image regions, two categories of agents named respectively darkening agents and lightning agents are defined. According to their category, the agents follow respectively the dark regions or the light regions. By remaining on crest lines, the agents strengthen the difference of contrast between the pairs of neighboring regions. The proposed system is well dedicated to images containing roof edges (detected by the discontinuity of the first derivative of the image). Indeed, theses edges characterize the considered images by the authors.

The previous multi-agent approaches like most of edge based approches, have proceeded to detection without any region representation. This does not facilitate the region based segmentation of the treated images.

In the category of region based methods, Liu et al. [7] have used a reactive agent based system for the segmentation of Magnetic Resonance Images (MRI) of the brain. The authors used four types of agents, which correspond to the four tissues in the pathological cerebral matter. A first generation of agents is initialized inside the various tissues. Then, when every agent recognizes its region, it creates offspring agents and places them so that they are lucky to find more other homogeneous pixels. For the same type of images, Richard and his co-authors [9] proposed a multi-agent system were agents are organized hierarchically as follows: 1) One global control agent which partition the image volume in partial volumes; 2) Local control agents, allocated each one to a partial volume; 3) tissue dedicated agents, which work under the control of a local control agent, to segment the tissue inside a partial volume. So, the detection is performed at the lowest level by the tissue dedicated agents, then synthetized by the local control agents and the global control agent.

Like most of the region based approaches of image segmentation, the previous two works follow supervised approaches, where it is necessary to know the number of regions and their shapes. We introduce into this paper a new approach which is general and unsupervised for range image segmentation. We show that the competitive alignment of edges performed by reactive agents allows the emergence of a collective segmentation of an image. The proposed approach is unsupervised. It makes no preliminary assumption on the number or the nature of the regions which form the images to be segmented.

3 Image Segmentation by Edge Emergence

In order to do not make any assumption on the shape of regions, we have used the directional curvatures of surface, as the homogeneity criterion. Let a pixel (x, y), the two directional curvatures, respectively horizonal C_h and vertical C_v, are defined as follows:

$$C_h(x, Y) = -\frac{I''(x, .)}{1 + I'^2(x, .)} \tag{1}$$

$$C_v(X, y) = -\frac{I''(., y)}{1 + I'^2(., y)} \tag{2}$$

where $I'(x, .)$ and $I''(x, .)$ are the first and the second derivative of the image following the x axis, by fixing the value of y to Y. The same, $I'(., y)$ and $I''(., y)$ are the first and the second derivative of the image following the y axis, by fixing the value of x to X. C_h expresses the curvature projection of the surface of the plane $y = Y$, while C_v expresses the curvature projection of the surface on the plane $x = X$. A contiguous set R of pixels $\{(x_k, y_k), k \in R\}$ is considered homogenous, and represents an image region, if all of its pixels have, according to a given threshold Tr_c, the same directional curvatures:

$$\forall k, l \in R; |C_h(x_k, y_k) - C_h(x_l, y_l)| < Tr_c \wedge |C_v(x_k, y_k) - C_v(x_l, y_l)| < Tr_c$$

Tr_c is a curvature comparison threshold where the value is automatically set at the stage of parameter learning (see Section 4). So, the pixels on interest, which are situated at the borders of the true regions or are in the noise regions, are detected when one of the two curvature C_h ou C_v changes.

3.1 System Dynamic

A high number of reactive agents (N=3500 see Section 4) are initialized at random positions in the image. After, agents start to move within the image following random directions. An agent searchs for an homogenous region around its current position (x_c, y_c), by checking the last L visited pixels. The path length in pixels L (set at Section 4) allows the agent to be sure that it is within a homogenous region. If these last L pixels have all the same directional curvatures, the agent considers that it is within a homogenous region ($CurrentRegion$). It acquires then the ability to alter the image ($AlterationAbility \leftarrow$ TRUE). This ability allows the agent to modify the depths of the encountered pixels. In its future moves, it smoothes the encountered pixels if these latter belong to the surface on which it currently moves. When the agent arrives on the first non homogenous pixel, it aligns this pixel to the current surface. The agent loses its alteration ability and restarts to search for a new homogenous region. The next algorithm introduces the method "*step()*" executed by an agent at every step on its path.

```
Initialisation :
  $AlterationAbility \leftarrow$ False
  $(x_c, y_c) \leftarrow (random(widthImg - 1), random(heightImg - 1))$
  $l \leftarrow 0$ // Set the path length
```

3.2 Noise Region Erasing and Edge Emergence

A noise region in a range image is either a homogenous region with weak size, or a non homogenous region formed by noise pixels having random or aberrant

Algorithm 1. Method *step()*

```
$(x_c, y_c) \leftarrow moveOnNeighbourPixel()$
if NOT $AlterationAblilty$ then
  if $NeighbourhoodHomogeous(x_c, y_c)$ then
    $l \leftarrow l + 1$
    if $l \geq L$ then
      $AlterationAbility \leftarrow$ TRUE
      $CurrentRegion \leftarrow region(x_c, y_c)$
    end if
  end if
else
  if $Belong(x_c, y_c, CurrentRegion)$ then
    $SmoothPixel(x_c, y_c)$
  else
    $alignPixel(x_c, y_c, CurrentRegion)$
    $AlterationAbility \leftarrow$ FALSE
    $l \leftarrow 0$
  end if
end if
```

depths. In the case of a weak size, when it is crossed by an agent, this latter can not cover L homogenous contiguous pixels. When the region is formed only by noise pixels, an agent when crossing it does not find at least L homogenous pixels to be able to initialize a homogenous region whichever the diameter of the region. These regions are progressively erased by aligning their border pixels to the surrounding true regions. Indeed, when an agent comes on a pixel on the border of a noise region, it aligns this pixel and goes in this region. Since it could not cross L contiguous homogenous pixels, it remains incapable to alter pixels within the noise region. When it leaves this region, it does not align the first encountred pixel in the homogenous surrounding region. So, the border of the noise regions is then continually aligned from outside to the surrounding true regions. The noise regions will disappear after several times agents cross theses regions. Note that one noise region is surrounded by agents of a same group. Agents of this group are those moving on the true region which surrounds the noise region.

On the other hand, a segment of a thick edge, situated between two homogenous regions, is surrounded by two groups of agents. Agents of each group are situated entirely within only one region. Agents which cross the segment align the pixels of the segment border to the region from where they come. The edge segment between the two homogenous regions is thus continually thinned by the two groups of agents (Fig. 1a and 1c).

When the edge becomes thin (one pixel wide), the agents of the two groups will become in competition to align the pixels of the thinned edge. Indeed, the pixels aligned by the agents of a giver group, let A in Fig 1b are immediately realigned

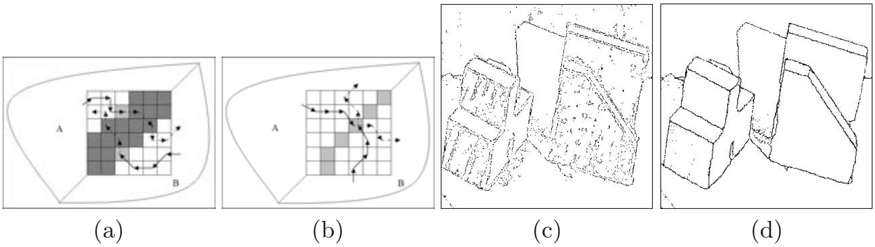


Fig. 1. Edge thinning (image abw.test.6). (a),(c) Edge pixels at t=800 ; (b),(d) Edge pixels at t= 8000.

to the second region (B) by the agents of the second group. So, the pixels of the edge are continually switched between the two regions. Consequently, whichever the number of alignments of these pixels, they remain emergent in the image (Fig. 1d). Note that this result is not coded in any agent, but it emerges from the collective action of all the agents in the image.

At the end of the process, noise regions are erased, and edges are thinned. So, a simple region growing controlled by the detected edges allows to produce a region based segmentation of the image.

4 Experimentation

We have used the framework proposed by Hoover et al. [3] to evaluate our approach, with range images from the set K2T containing both curved and polyhedral objects. All the images have a size of 640 × 480 pixels. A set of 10 range images was used to select the values of the used parameters: the number of agents N=3500; the path length L=7; and the curvature threshold Tr_c. The selected value correspond to the best segmentation, expressed as the number of the regions correctly detected, according to a ground truth (GT) segmentation [3].

In order to show how the noise regions are progressively erased and how edges are thinned, we show aligned pixels at regular intervals of time t. Figure 2a shows a sample of a range image. Figures 2b, 2c, 2d, 2e and 2f show the aligned pixels respectively at t=500,3500,7500,10500 and 13500. As the time progresses edges are detected and thinned, and noise regions are erased.

Fig. 3 shows the average number of instances of correct detection corresponding respectively to our method (DCIS for Distributed and Collective Image Segmentation) and the method of EG (Edge based Segmentation), where authors have proposed a segmentation method for curved objects [4,5]. Theses results are obtained with the set K2T of images containing curved objets, and are computed according to a compare tool tolerance T, which is used to express the comparison tolerance. We can observe that our method records scores better than those of EG for T in {50%, 60%, 70% and 80%}. The two methods record equivalent scores for T in {90%,95%}.

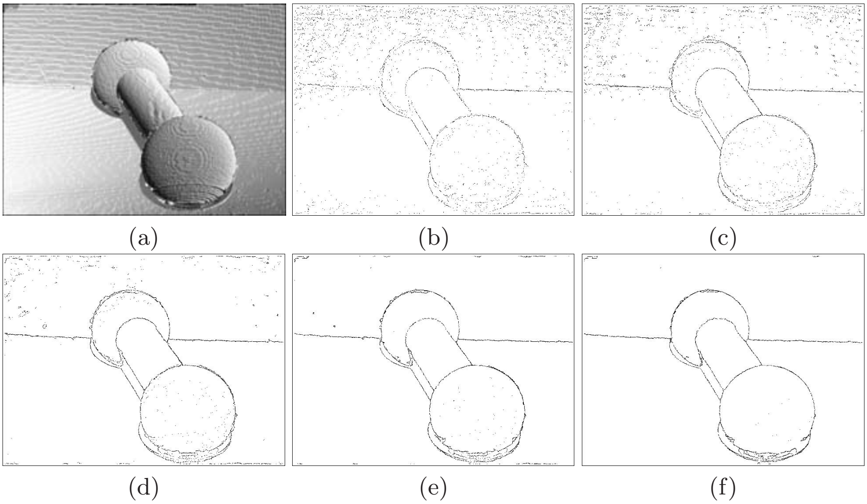


Fig. 2. Segmentation Progression. (a) Rendered Range Image; (b) at t=500, (c) at t=3500 ; (d) at t=7500 ; (e) at t=10500 ; (f) at t=13500.

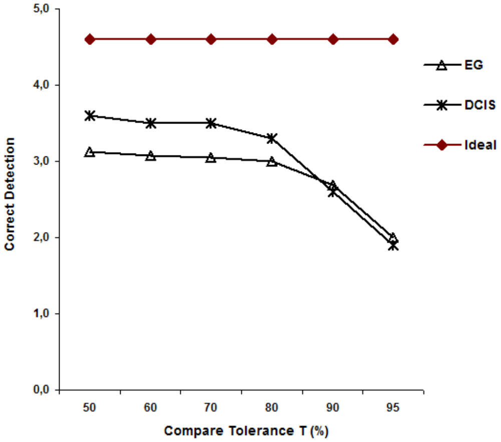


Fig. 3. Average number of correct detections in the set K2T for the two methods EG and DCIS according to T ; $50\% \leq T < 100\%$

5 Conclusion

We have proposed in this paper a new approach for edge detection and noise erasing in range images. This has allowed the segmentation of an image in its several homogenous regions. Combining the surface curvature and the competitive alignment of pixels has allowed to produce an accurate segmentation of images containing curved objects. These latter are recognized as difficult to segment in the case of range images. According to our approach, the detection of an edge segment results from the alternative alignment of its pixels. This alignment is performed by agents coming from the two regions. On the other hand, noise regions are progressively erased by the continuous alignment of the pixels of their borders to the homogenous surrounding regions. We think that the proposed approach could be applied to other types of images. For this, it's necessary to define a homogeneity criterion of regions in treated images.

References

1. Ballet, P., Rodin, V., Tisseau, J.: Edge detection using a multiagent system. In: 10th Scandinavian Conference on Image Analysis, Lapeenranta, Finland, pp. 621–626 (1997)
2. Ferber, J.: Les systèmes multi-agents : vers une intelligence collective. Informatique, Intelligence Artificielle. Inter Éditions (1995)
3. Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K.W., Eggert, D.W., Fitzgibbon, A.W., Fisher, R.B.: An experimental comparison of range image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(7), 673–689 (1996)
4. Jiang, X., Bunke, H.: Edge detection in range images based on Scan Line approximation. Computer Vision and Image Understanding 73(2), 183–199 (1999)
5. Jiang, X.: Recent advances in range image segmentation. In: Selected Papers from the International Workshop on Sensor Based Intelligent Robots, London, UK, pp. 272–286. Springer, Heidelberg (1999)
6. Jones, J., Saeed, M.: Image enhancement, an emergent pattern formation approach via decentralised multi-agent systems. Multiagent and Grid Systems Journal (ISO Press) Special Issue on Nature inspired systems for parallel, asynchronous and decentralised environments 3(1), 105–140 (2007)
7. Liu, J., Tang, Y.Y.: Adaptive image segmentation with distributed behavior-based agents. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(6), 544–551 (1999)
8. Mazouzi, S., Batouche, M.: A new bayesian method for range image segmentation. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 453–466. Springer, Heidelberg (2007)
9. Richard, N., Dojat, M., Garbay, C.: Automated segmentation of human brain MR images using a multi-agent approach. Artificial Intelligence in Medicine 30(2), 153–176 (2004)
10. Rodin, V., Benzinou, A., Guillaud, A., Ballet, P., Harrouet, F., Tisseau, J., Le Bihan, J.: An immune oriented multi-agent system for biological image processing. Pattern Recognition 37(4), 631–645 (2004)

Learning an Efficient Texture Model by Supervised Nonlinear Dimensionality Reduction Methods

Elnaz Barshan, Mina Behravan, and Zohreh Azimifar

School of Electrical and Computer Engineering,
Shiraz University, Shiraz, Iran
barshan@cse.shirazu.ac.ir

Abstract. This work investigates the problem of texture recognition under varying lighting and viewing conditions. One of the most successful approaches for handling this problem is to focus on textons, describing local properties of textures. Leung and Malik [1] introduced the framework of this approach which was followed by other researchers who tried to address its limitations such as high dimensionality of textons and feature histograms as well as poor classification of a single image under known conditions.

In this paper, we overcome the above-mentioned drawbacks by use of recently introduced supervised nonlinear dimensionality reduction methods. These methods provide us with an embedding which describes data instances from the same classes more closely to each other while separating data from different classes as much as possible. Here, we take advantage of the superiority of modified methods such as "Colored Maximum Variance Unfolding" as one of the most efficient heuristics for supervised dimensionality reduction.

The CUReT (Columbia-Utrecht Reflectance and Texture) database is used for evaluation of the proposed method. Experimental results indicate that the algorithm we have put forward intelligibly outperforms the existing methods. In addition, we show that intrinsic dimensionality of data is much less than the number of measurements available for each item. In this manner, we can practically analyze high dimensional data and get the benefits of data visualization.

Keywords: Texture Recognition, Texton, Dimensionality Reduction.

1 Introduction

Texture is a fundamental characteristic of natural materials and has the capacity to provide important information about scene interpretation. Consequently, texture analysis plays an important role both in computer vision and in pattern recognition. Over the past decades, a significant body of literature has been devoted to texture recognition based on mainly over-simplified datasets. Recently, more and more attention has been paid to the problem of analyzing textures achieved in different illumination and viewing directions. As Figure 1 shows, recognizing textures with such variations generally causes much trouble. Leung and Malik [1] were amongst the first to comprehensively study such variations. They proposed 3-D textons which are cluster centers of a number of predefined filter responses (textons) over a stack of images with different viewpoint and lighting conditions. The basic idea here is to build a universal vocabulary from these

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 209–216, 2009.

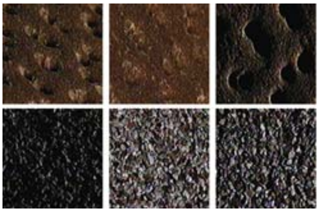


Fig. 1. Changing viewpoint and illumination can have a dramatic impact on the appearance of a texture image. Each row shows texture images of the same class under different viewpoint and lighting conditions.

textons describing generic local features of texture surfaces. Given a training texture class, the histogram of its 3-D textons forms the model corresponding to that texture. In the training stage, the authors acquired a model for each material using stacked images of different albeit a priori known conditions. This model, however, requires the test images to be in the same order as in the training. Leung and Malik also developed an algorithm for classifying a single image under known conditions. Yet, this method does not classify a single image as efficient as the case for the multiple images. Later, Varma and Zisserman [2] presented an algorithm based on Leung and Malik's framework, without requiring any a prior knowledge of the imaging conditions. In Varma and Zisserman's method, textons are obtained from multiple unregistered images of a particular texture class using K-means clustering. A representing model for each class is brought out using the texture library which is, actually, a collection of textons from different texture classes.

For the purpose of achieving a faithful representation of various textures, a finite set of textons (i.e., texton library) closely representing all possible local structures, ought to be obtained. Hence, the cardinality of our texton library must be considerably large. Nevertheless, this may, by itself, cause high-dimensional models. To address this issue, Cula and Dana [3] employed the method of Principal Component Analysis (PCA) and compressed the feature histogram space into a low-dimensional one. Applying PCA as a method for unsupervised linear dimensionality reduction causes a number of limitations to be discussed later on.

In this paper, we focus on the problem of high dimensionality of texture models and, furthermore, introduce an efficient algorithm for classifying a *single* texture image under *unknown* imaging conditions. Here, our attempt is to shed light on a new approach to overcome this difficulty. In this viewpoint of ours, a richer space is sought after that can reflect modes of variability which are of particular interest. As a result, we propose to project data onto an ideal space peculiar to our problem. Not only is the new space thus gained supposed to be of low dimension, but also it has to provide us with a better representation of data, i.e. to be more discriminative for the classification algorithm. In other words, we aim at transforming the ill-posedness of texture classification into a better-posed problem. To find this transformation, we take the benefits of recently introduced supervised nonlinear dimensionality reduction methods.

The rest of this paper is organized as follows: Section 2 provides an overview on texton-based texture representation. Next, we briefly describe one of the most efficient heuristics for reducing the dimensionality of nonlinear data. In section 3, we introduce our new method which represents a model of enough capability for classifying a single image under unknown imaging conditions. Experimental results of the proposed algorithm are presented in section 4 followed by “Conclusion” in section 5.

2 Background Review

2.1 Texton-Based Texture Representation

A texture image is constructed based on certain structures, such as spot-like features of various sizes, edges with different orientations and scales, bar-like characteristics and so on. It was reported that local structure of a texture can be closely represented by its responses to an appropriate filter bank [4,5,6].

Different filter banks focus on different constructive structures. Accordingly, Leung and Malik [1] introduced an appropriate LM filter bank which was later employed by a number of other researchers. The LM set is a multi-scale, multi-resolution filter bank that has a combination of edge, bar and spot filters. It consists of the first and the second derivatives of Gaussian (at six orientations and three scales), eight Laplacian of Gaussian (LOG) filters and four Gaussian filters, a total of 48 filters. In this study we used this filter bank.

One of the fundamental properties of textures is pattern repetition, which means that filter responses to only a small portion of texture image are sufficient to describe its structure. This small set of prototype response vectors of one image was called 2-D textons by Leung and Malik. They also proposed 3-D textons definition; this definition is based on the idea that the vectors obtained from concatenating filter responses of different images of the same class will encode the appearance of dominant features in all of the images. They used 3-D textons to represent a framework for recognizing textures under different imaging conditions. Since the inspiration of our work comes from the Leung and Malik’s algorithm [1], let us briefly review this method.

The Leung and Malik’s algorithm uses 3-D textons from all the texture classes to compute a universal vocabulary. To construct such a desirable vocabulary, the K-means clustering algorithm is applied to the data from each class individually. The class centers are, then, merged together to produce a dictionary. This dictionary should be pruned in order to produce a more efficient, faithful and least redundant second version. After constructing the vocabulary, different images from each class are passed through the filter bank and stored in a large vector, which is then assigned to the nearest texton labels from the said dictionary. The histogram of texton frequencies is computed in such a manner as to obtain one model per class. Textons and texture models are learnt from training images. Once this is done, classification of a test image is done by computing a model from images with different imaging conditions, as in the training stage. The algorithm selects the class for which chi-square distance between the sample histogram and the model histogram could be minimized. Readers interested in other aspects of the original algorithm are referred to Leung and Malik’s original paper [1]. Despite the fact that Leung and Malik’s algorithm has numerous advantages, it has its own limitations discussed by several authors from different aspects. These disadvantages were to be addressed by the very authors. In section 3, we discuss shortcomings of this algorithm and introduce a new approach based on dimensionality reduction methods.

2.2 Dimensionality Reduction

The problem of dimensionality reduction and manifold learning has recently attracted much attention on the part of many researchers. Manifold learning is a method to retrieve low dimensional global coordinates that faithfully represent the embedded manifold in the high dimensional observation space.

Most dimensionality reduction methods are unsupervised. That is to say, they do not respect the label or the real-valued target covariate. Therefore, it is not possible to guide the algorithm towards those modes of variability that are of particular interest. For example, where possible, by using labels of a subset of the data according to the kind of variability that one is interested in, the algorithm can be guided to reflect this kind of variability.

Amongst the proposed supervised nonlinear dimensionality reduction methods, "Colored Maximum Variance Unfolding" (CMVU) [7] is of much interest and capability.This method is built upon "Maximum Variance Unfolding" (MVU) method [8]. By integrating two sources of information, data and side information, CMVU is able to find an embedding which: 1) preserves the local distances between neighboring observations, and 2) maximally aligns with the second source of information (side information). Theoretically speaking, CMVU constructs a kernel matrix **K** for the dimension-reduced data X which has the capacity to keep the local distance structure of the original data Z unchanged, so that X maximally depends on the side information Y as described by its kernel matrix **L**. This method is formulated by the following optimization problem:

$$\boxed{\begin{array}{l}\text{Maximize tr HKHL subject to:}\\ 1. K \succeq 0\\ 2. K_{ii} + K_{jj} - 2K_{ij} = d_{ij} \text{ for all } (i,j) \text{ with } \eta_{ij} = 1\end{array}}$$

where $K, L \in \mathbb{R}^{m*m}$ are the kernel matrices for the data and the labels, respectively, $H_{ij} = \delta_{ij} - m^{-1}$ centers the data and the labels in the feature space, and binary parameter η_{ij} denotes whether inputs z_i and z_j are k-nearest neighbors or not. The objective function is an empirical estimate of "Hilbert-Schmidt Independence Criterion" (HSIC) that measures the dependency between data and side information [9]. This optimization problem is an instance of semi-definite programming (SDP). From the solution of SDP in the kernel matrix K, output points X_i could be derived using singular value decomposition. Figure 2 illustrates embedding of 2007 USPS digits produced by CMVU and PCA, respectively.

3 Methodology

In this section, we discuss different texton-based texture representation methods. Then, we present our new method to address all accompanying drawbacks, and will show its superiority compared to other methods each focusing on a specific limitation.

As stated in the previous section, issues associated with the use of 3D textons to classify 3D texture images are:

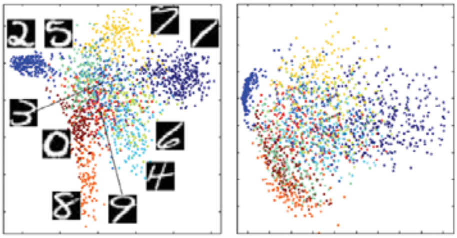


Fig. 2. Embedding of 2007 USPS digits produced by CMVU and PCA, respectively [7]

- increased dimensionality of feature space to be clustered in the later stages,
- increased time complexity of the iterative procedure to classify a single image which causes the convergence problem,
- necessity of a set of ordered texture images captured under known imaging conditions, and
- introduction of only one comprehensive model per class whereas it is unlikely that a single model can fully account for the various appearance of real-world surfaces.

By use of 2-D textons, none of the above problems would appear. 2-D textons are cluster centers of filter responses over a single image (and not over a stack of images) captured at different conditions. The problem here is how we should represent different instances from the same class as being inter-related while preserving the between-class distances. One solution is to select the models which best represent their texture classes. Cula and Dana [3] proposed a model selection algorithm in a low dimensional space. They fitted a manifold to low dimensional representation of models specifically generated for each class and removed the models which least affected the manifold shape. Their algorithm, notwithstanding, introduces some drawbacks. For projecting models into a low dimensional space, they utilized PCA which is an unsupervised linear dimensionality reduction method. The PCA works well if the most important modes of data variability are linear. But in this study, the variability of models cannot be expressed linearly and this causes poor performance of PCA. The second problem stems from the fact that two different distance measures are used in constructing the manifold path in the training stage and selecting the closest model in the classification stage. In other words, when constructing the manifold path, at each step the closest point in terms of imaging angles is chosen, while in classification phase, the closest surface class is selected in terms of distance between models feature vectors. Another significant issue is that this algorithm ignores inter-class variation between textures since the models for a texture are selected without considering the other texture classes.

Having discussed the above issue, we propose to analyze this problem from another viewpoint: *reducing the dimensionality of model histograms to their intrinsic dimensionality*. By mapping the models to a very low dimensional space, the complexity of the classification decreases and model selection can take the benefits of data visualization. It is important to note that the basic modes of variability of our data are nonlinear. Therefore, the dimensionality reduction method should be capable of unfolding the manifold on which the nonlinear dataset is lying. On the other hand, we are searching for a space in which models from the same classes stay more closely while models from different classes remain as much discriminated as possible.

Here we take the advantages of CMVU, which is one of the most efficient heuristics for supervised dimensionality reduction, as discussed in § 2. This method generates brilliant results for training data, e.g., it is empirically observed that the most significant modes of the variability of a dataset with dimensionality of 1200 can be presented in a space of as low as five dimensions. It confirms our reasoning of selecting the CMVU to visualize the train data. This method, however, faces some complications in projecting the test data. Desired embedding for training data could be computed with respect to its labels. Because of the fact that at the testing time the second source of information (the labels) is not available, this method does not provide us with an embedding of testing data to the space in which the training data is embedded. Herein, we choose to project the testing data based on the fundamental idea of "Locally Linear Embedding" (LLE) [10]. The projection procedure for testing data S is as follows:

Alg. 1. The projection procedure for testing data

```
Input: training data matrix in the original space, Z, projected training data matrix, X, testing data matrix in the original space, S, and the number of testing data, $m$
Onput: Projected testing data matrix, P
1: for all $i \in \{1 \ldots m\}$
2:     $N = \{z_j \in Z | \eta_{ij} = 1\}$
3:     $W = \text{argmin}_W E(W) = |s_i - \Sigma_j w_{ij} N_j|$
4:     $p_i = \Sigma_j W_{ij} x_j$
5: end for
```

The projection of testing data using this LLE-like method causes some negligible differences under the circumstances of the presence of labels being computed using CMVU.

4 Experimental Results

We perform all experiments on the CUReT dataset [11]. This dataset provides a starting point in empirical studies of texture surfaces under different viewing and illumination directions. This database contains 61 different textures, each observed with over 200 combinations of viewpoint and illumination conditions.

In order to construct our texton library, 40 unregistered images with different imaging conditions from 20 texture classes are employed. We use the same texture set as the one examined by Cula and Dana [3]. We extract 2-D textons from each image, and apply K-means algorithm to the texture classes individually in order to obtain 60 centers from each different materials. These 1200 centers are used as initial points for the final clustering step, which produce an efficient and least redundant dictionary from 1200 textons.

To justify the effectiveness of our approach, we have performed three sets of experiments. 10 arbitrary texture images from each texture class are selected in all three of experiments. Thence, the total number of test images is 200. In the first experiment,

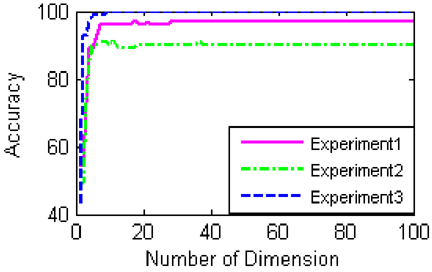
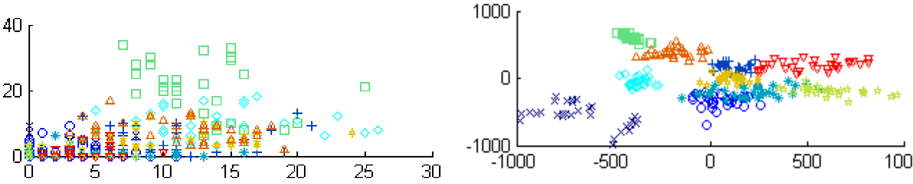


Fig. 3. Classification rate on CURet dataset for different projection dimensions. Three sets of experiments have been performed. In experiment (1) exactly the same images involved in constructing the vocabulary have been used. Experiment (2) is a bit more complex, in the sense that testing image conditions differ from those used in constructing the vocabulary. In Experiment (3) two disjoint sets of texture classes are used in library construction and the texture recognition, separately.



(a) Original Space

(b)CMVU Space

Fig. 4. The first two dimensions of CUReT dataset in the original space and the space produced by CMVU, respectively. Dot shapes are used to denote textures from different classes.

exactly the same images involved in constructing the vocabulary have been used. The second experiment is a bit more complex, in the sense that testing image conditions differ from those used in constructing the vocabulary. In the last experiment, the most complex one, two disjoint sets of texture classes are used in library construction and texture recognition, separately. Figure 3 shows the percentage of correctly classified test images as a function of dimensions used to represent projected models by CMVU. This figure clearly shows that up to a certain level the accuracy increases with dimensionality and converges to a fixed point with very low variability. Additionally, this Figure shows better results for experiment 3, which is the consequence of selecting more discriminative classes than the other sets chosen for constructing the library. In Figure 4 the first two dimensions of data in the original space is shown as well as its projection in/on to the new space using CMVU. Obviously enough, CMVU introduces a clear data separation with an excellent visualization.

5 Conclusions

This paper introduces the idea of supervised nonlinear dimensionality reduction to alleviate the difficulties associated with texture recognition. Although we were not the first to address the high dimensionality of texture models, the contribution of this work is its efficient mapping of data nonlinearity, i.e., we have shown how to represent the data intrinsic information while magnifying the descriptive properties of the original feature space. Besides, we proposed a LLE-like approach to cope with shortcoming of CMVU in projecting the test data when carrying no side information. This paper presents a new framework to efficiently visualize a hugely dimensioned data in a very low dimension yet rich space.

This study can be extended in different directions: 1) orientation and scale invariant features may be extracted using techniques such as gradient histograms, 2) advanced classifiers and clustering algorithm can be investigated, and 3) the data visualization techniques may also be employed in selecting the most discriminative texture models.

References

1. Leung, T.K., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision 43(1), 29–44 (2001)
2. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision 62(1)
3. Cula, O.G., Dana, K.J.: 3d texture recognition using bidirectional feature histograms. International Journal of Computer Vision 59(1), 33–60 (2004)
4. Clark, M., Bovik, A.C., Geisler, W.S.: Multichannel texture analysis using localized spatial filters. IEEE Trans. Pattern Anal. Mach. Intell. 12(1), 55–73 (1990)
5. Randen, T., Husoy, J.H.: Filtering for texture classification: A comparative study. IEEE Trans. Pattern Anal. Mach. Intell. 21(4), 291–310 (1999)
6. Prabhakar, S., Jain, A.K., Hong, L.: A multichannel approach to fingerprint classification. IEEE Trans. Pattern Anal. Mach. Intell. 21(4), 348–359 (1999)
7. Smola, A.J., Borgwardt, K.M., Song, L., Gretton, A.: Colored maximum variance unfolding. In: NIPS (2007)
8. Weinberger, K.Q., Saul, L.K.: An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In: AAAI (2006)
9. Bousquet, O., Smola, A.J., Gretton, A., Scholköpf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: ALT, pp. 63–77 (2005)
10. Saul, L.K., Roweis, S.T.: Think globally, fit locally: Unsupervised learning of low dimensional manifold. Journal of Machine Learning Research 14, 119–155 (2003)
11. Nayar, S.K., Koenderink, J.J., Dana, K.J., Ginneken, B.v.: Reflectance and texture of realworld surfaces. ACM Trans. Graph. 18(1), 1–34 (1999)

A Fuzzy Segmentation Method for Images of Heat-Emitting Objects

Anna Fabijańska

Department of Computer Engineering, Technical University of Lodz
18/22 Stefanowskiego Str., 90-924, Lodz, Poland
an_fab@kis.p.lodz.pl

Abstract. In this paper a problem of soft image segmentation is considered. An approach for segmenting images of heat-emitting specimens is introduced. Proposed algorithm is an extension of fuzzy C-means (FCM) clustering method. Results of applying the algorithm to exemplary images of heat-emitting specimens are presented and discussed. Moreover the comparison with results of standard fuzzy C-means clustering is provided.

Keywords: image segmentation, fuzzy sets, clustering methods, FCM, high-temperature measurement, surface property of metal.

1 Introduction

Image segmentation is an essential task in machine vision applications. It addresses the problem of partitioning the image into disjoint regions according to the specific features (gray levels, texture etc.) [1]. Different approaches to image segmentation have been proposed in the literature. The most popular are histogram-based methods [1][2], region growing approaches [1][3], edge-based methods [1][4], clustering techniques [5] and watershed segmentation [1][6]. However most of these methods are hard techniques which provide crisp segmentation of images by qualifying each pixel to the unique region.

Hard segmentation techniques are often insufficient for practical applications of vision systems. Crisp partitioning of the image is often inaccurate and erroneous. Therefore the growth of interest in soft segmentation techniques can be observed recently. They are based on fuzzy set theory [7][8] end extract fuzzy regions (subsets of pixels) from the fuzzy image. In soft segmentation approaches each pixel can be qualified into multiple regions with different degree of membership [7][9][10].

2 The Experimental Set Up

Images considered in this paper were obtained from computerized system for high-temperature measurements of surface properties of metals and alloys. The system “Thermo-Wet” determines wetting angles and surface tension of liquid materials up to 1800^0C. The description of “Thermo-Wet” architecture can be found in [11][12].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 217–224, 2009.

The considered system applies sessile drop method [13] to calculate surface parameters. Surface tension and contact angles are calculated from images presenting melted specimens of investigated materials. Exemplary images obtained during the measurement process are presented in Figure 1. They are 8-bit monochromatic images of the resolution 320×240 pixels.

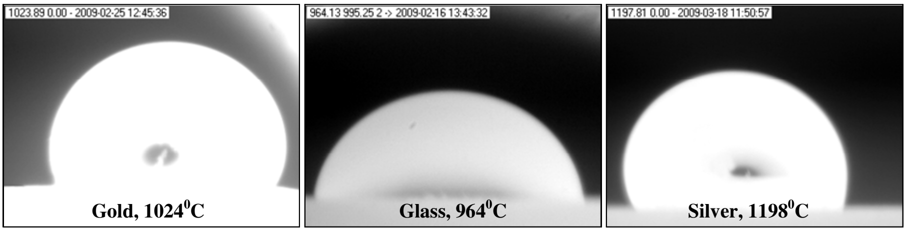


Fig. 1. Exemplary images obtained from "Thermo-Wet" vision system

After the image is acquired it is subjected to image segmentation which determines specimen shape and location of upper edge of the base-plate. Next, specimen shape analysis is carried out in order to determine characteristic geometric parameters of specimen (see Fig. 2). They are related to surface tension and contact angles through appropriate formulas arising from the sessile drop method [13]. Especially Porter's formula is applied in this stage. More detailed information about the measurement process is given in [11][12].

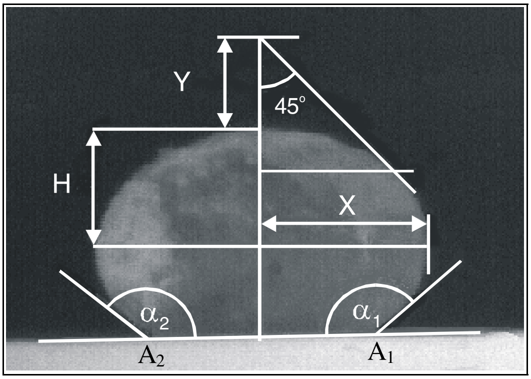


Fig. 2. The exemplary specimen with important geometric parameters marked

Image segmentation is crucial task for measurements of surface parameters. It determines specimen shape and location of upper edge of the base plate. However the segmentation of images of heat-emitting specimens is very challenging. Problems with segmentation are caused by "aura" i.e. glow that forms itself around the specimen. Aura significantly hinders accurate location of specimen edges. It blurs the border between the background and the specimen. Hard segmentation techniques join aura with the object which affect with specimen dimensions increase.

In this paper fuzzy approach to segmentation of images presenting heat-emitting specimens is introduced. The algorithm which uses an extension of fuzzy C-means (FCM) clustering method is proposed. It iteratively segments aura. In consequence aura is effectively excluded and specimen shape is preserved after segmentation.

A brief overview of FCM and detailed description of the proposed approach are given in the following sections.

3 Background

Fuzzy C-means (FCM) algorithm clusters pixels into a specified number of regions (clusters). It is based on minimization of the objective function J_m given by equation (1) [13][14].

$$J_m = \sum_{i\in\Omega}\sum_{j=1}^{C} u_{ij}^{m}\left\|\mathbf{x}_i - \mathbf{c}_j\right\|^2 . \tag{1}$$

where:

m - a real number greater than 1;
C - number of clusters;
$\mathbf{x}_i$ - vector of pixel properties at location i;
u_{ij} - the degree of membership of $\mathbf{x}_i$ in the j-th cluster;
$\mathbf{c}_j$ - centroid of the j-th cluster;
||•|| - norm expressing the distance in P-dimensional feature space;
Ω - set of pixels in the image domain.

Fuzzy clustering is carried out through an iterative minimization of the objective function J_m with the update of the degree of membership u_{ij} and the cluster centers $\mathbf{c}_j$ by equations (2) and (3) respectively.

$$u_{ij} = \sum_{k=1}^{C}\left(\frac{\left\|\mathbf{x}_i - \mathbf{c}_j\right\|}{\left\|\mathbf{x}_i - \mathbf{c}_k\right\|}\right)^{-\frac{2}{m-1}} . \tag{2}$$

$$\mathbf{c}_j = \frac{\sum_{i\in\Omega} u_{ij}^{m}\mathbf{x}_i}{\sum_{i\in\Omega} u_{ij}^{m}} . \tag{3}$$

The minimization is stopped when equation (4) gets fulfilled.

$$\max_{ij}\left\{\left|u_{ij}^{(s+1)} - u_{ij}^{(s)}\right|\right\} < \varepsilon . \tag{4}$$

Where ε is a termination criterion and s is the iteration step.

After minimization of the objective function is finished maximum-membership segmentation is usually applied. During this process pixels are classified into the cluster with the highest degree of membership.

4 Proposed Approach

4.1 Pixel Description

In case of analyzed images a pixel at location i is described by a vector of features $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}, x_{i7}]$ in 7-dimensional (P=7) feature space where:

x_{i1} - intensity of i-th pixel;
x_{i2} - an average intensity of $n\times n$ neighborhood of i-th pixel;
x_{i3} - standard deviation of intensity in $n\times n$ neighborhood of i-th pixel;
x_{i4} - gradient magnitude in i-th pixel;
x_{i5} - gradient direction in i-th pixel;
x_{i6} - an average gradient magnitude in $n\times n$ neighborhood of i-th pixel;
x_{i7} - an average gradient direction in $n\times n$ neighborhood of i-th pixel.

Neighborhood of size 3×3 pixels is considered (n=3). Sobel operator [1] is applied to determine magnitude and direction of gradient. The distance between pixels is computed using Euclidean metric (5).

$$\left\| \mathbf{x}_i - \mathbf{x}_j \right\| = \sqrt{\sum_{k=1}^{P} (x_{ik} - x_{jk})^2} \, . \tag{5}$$

Extending number of features describing a pixel to proposed number of elements increases quality of image segmentation. Tests proved that using standard features i.e. pixel intensity and standard deviation is insufficient to obtain high-quality results.

4.2 Algorithm Description

Proposed approach clusters pixels into two (C=2) regions: the background (k=1) and the object (k=2). Clusterization is performed iteratively. In the consecutive iterations both regions compete for pixels with similar membership to both clusters.

The main steps of the algorithm are as follows:

1. Centers $\mathbf{c}_k^{(s)}$ k={1, 2} of the clusters are determined among unclassified pixels $\hat{\Omega}$.
2. Objective function $J_m^{(s)}$ given by Equation (1) is minimized in accordance with Equations (2)-(4) for parameter m=3 (i.e. for each pixel $\mathbf{x}_i$ membership measures $u_{ik}^{(s)}(\mathbf{x}_i)$ in k clusters are computed using fuzzy C-means clustering algorithm).
3. Pixels are temporarily assigned to clusters with maximum membership measure in accordance with equation:

$$\forall_{i\in\hat{\Omega}} \; \tilde{\partial}(\mathbf{x}_i) = \arg\max_{j\in[1,\,C]}(u_{ij}^{(s)}) \, . \tag{6}$$

where: $\tilde{\partial}(\mathbf{x}_i)$ is temporal affiliation of the pixel $\mathbf{x}_i$.

4. Among pixels temporarily assigned to k-th cluster threshold $T_{uk}^{(s)}$ for membership in the cluster is computed. For threshold determination the ISODATA (*Iterative Self-Organizing Data Analysis Technique*) algorithm [14][16] is applied.

5. Pixels $\mathbf{x}_i$ are permanently classified into clusters in accordance with the equation:

$$\forall_{i \in \hat{\Omega}} \; u_{ik}^{(s)} \geq T_{uk}^{(s)} \Rightarrow \partial(\mathbf{x}_i) = k \,. \tag{7}$$

   where: $\partial(\mathbf{x}_i)$ is a final affiliation of the pixel $\mathbf{x}_i$.

   In this step only pixels having the membership higher than the selected thresholds are qualified into to the regions (the object and the background). The remaining pixels are left unclassified. They are considered in the next iteration.
6. Steps 1-5 are repeated (for s=s+1) until all pixels are classified i.e. $\hat{\Omega}=\emptyset$.

Crucial problem for the algorithm is selection of cluster centers. In the proposed approach for the first iteration the cluster centers are set manually using the knowledge about characteristic features of analyzed images. The initial cluster centers represent ideal pixel belonging to the background $\mathbf{c}_1^{(1)}$=[0,0,0,0,0,0,0] and ideal pixel belonging to the object $\mathbf{c}_2^{(1)}$=[255,255,0,0,0,0,0]. In the following iterations one of the cluster centers is left unchanged and the second one is selected randomly from unclassified pixels.

For cluster centers selection bump-hunting algorithm [17] can be also used with the success. However its application increases time complexity of the proposed segmentation method.

5 Results

The method was extensively tested on images over a wide range of temperatures and strength of an “aura”. Results of applying the proposed image segmentation algorithm to the exemplary images of heat-emitting specimens are presented in Figure 3. Moreover the figure shows a comparison with results obtained by maximum membership fuzzy C-means segmentation and *the ground truths*. First column presents original images. The material of the specimen and its temperature is indicated on the each image. Images with different strength of an “aura” are considered. In the second column results of image segmentation obtained using the proposed method are shown. The third column presents results of an ordinary (i.e. maximum membership) fuzzy C-means clustering. In case of both considered segmentation methods pixels were described by vectors of features as described in Section 4.1. In the last column *the ground truths* are presented. They were obtained by manual segmentation performed by the skilled operator of the “Thermo-Wet” system.

Quantitative assessment of segmentation quality is presented in Table 1. Images presented in Figure 3 are considered.

Results obtained by the proposed method and maximum membership fuzzy C-means clustering (MM FCM) are compared with *the ground truths* by means of:

- *good matches* i.e. pixels properly qualified into both the object and the background;
- *false positives* i.e. background pixels qualified into the object;
- *false negatives* i.e. object pixels qualified into the background.

The material of the specimen and temperature describing the considered image are indicated in the column caption.

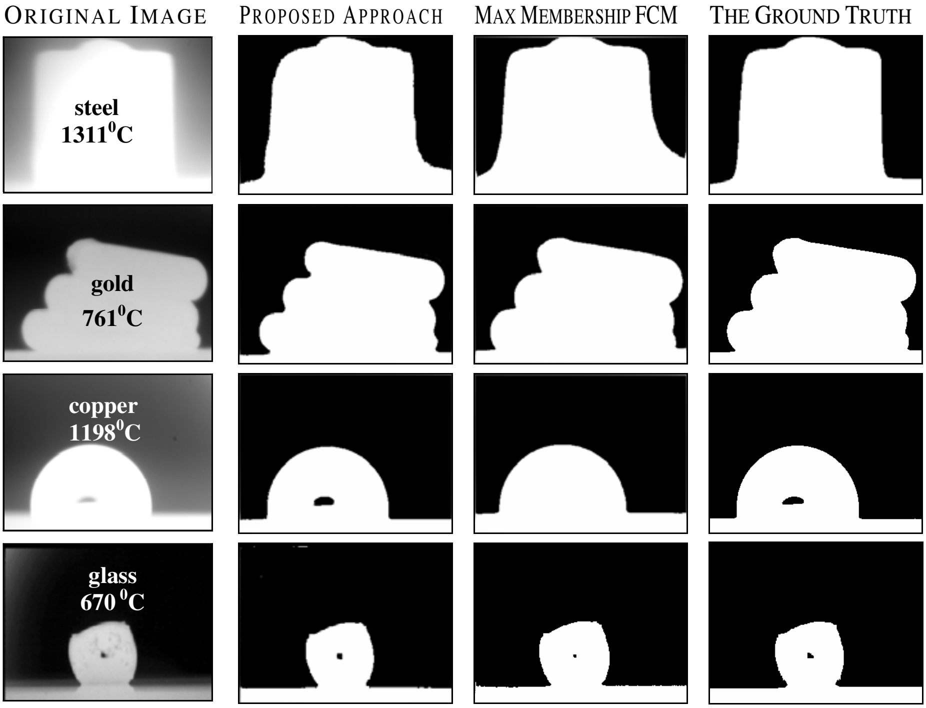


Fig. 3. Results of the proposed image segmentation algorithm compared with results obtained using an oridinary (i.e. maximum-membership) fuzzy C-means clustering and *the ground truths*. The type of the image is indicated over each column.

Table 1. Quantitative assessment of image segmentation quality

	Steel, 1311^{0}C		Gold, 761^{0}C		Copper, 198^{0}C		Glass, 670^{0}C	
	New method	MM FCM	New method	MM FCM	New method	MM FCM	New method	MM FCM
Good matches	96.38%	92.86%	98.41%	97.10%	99.52%	96.81%	99.36%	98.97%
False positives	2.92%	7.10%	1.58%	0.00%	0.03%	3.19%	0.63%	0.47 %
False negatives	0.70 %	0.04 %	0.01 %	2.90%	0.45 %	0.00 %	0.01 %	0.56 %

6 Discussion

Results presented in Figure 3 and Table 1 show clearly that the proposed extension of a fuzzy C-means segmentation algorithm efficiently segments images of heat-emitting specimens. High quality results are obtained for images of different intensity and strength of an "aura". Shape of objects after segmentation is well defined. Obtained contours are smooth, regular and free from defects. Moreover, important details of specimen shape are properly extracted.

Both the visual and the quantitative comparison with results obtained by maximum-membership fuzzy C-means segmentation prove that the proposed approach is much more accurate in segmenting an "aura". While traditional FCM joins aura with the object and significantly increases its dimensions, the new algorithm preserves specimen dimensions by qualifying an “aura” to the background. The difference between results obtained by both considered methods can especially be observed in case of images with significant aura.

Tests proved that in case of analyzed class of images the proposed segmentation algorithm finishes in less than ten iterations.

7 Conclusions

In this paper problem of fuzzy image segmentation was considered. Especially an extension of fuzzy C-means segmentation algorithm was introduced. The algorithm does not perform crisp classification of pixels into clusters with maximum membership measure but makes background and object compete for pixels in consecutive iterations.

Presented results prove that the proposed algorithm performs accurate segmentation of images of heat-emitting specimens. Specimen shape after segmentation is well defined - much better than in case of an ordinary FCM clusterization. Segmented objects are characterized by smooth and regular borders. Moreover an “aura” (i.e. glow that forms itself around the specimen) is effectively removed by the new approach. Traditional FCM approach joins aura with the object what increases object dimensions. Quality of obtained results is sufficient for further quantitative analysis of specimen shape.

Although the algorithm has been developed for a certain class of images, it can be successfully applied in a wide spectrum of applications as it does not take the advantage of knowledge about image properties.

Acknowledgements

The author is a scholarship holder of project entitled "Innovative education ..." supported by European Social Fund.

References

1. Gonzalez, R., Woods, E.: Image Processing. Prentice Hall, New Jersey (2007)
2. Liu, X., Wang, D.: Image and Texture Segmentation Using Local Spectral Histograms. IEEE Trans. Image Proc. 15(10), 3066–3077 (2006)
3. Fan, J., Zengb, G., Bodyc, M., Hacidc, M.: Seeded region growing: an extensive and comparative study. Pattern Recognition Letters 26(8), 1139–1156 (2005)
4. Silva, L., Bellon, O., Gotardo, P.: Edge-based image segmentation using curvature sign maps from reflectance and range images. In: IEEE Int. Conf. Image Processing, vol. 1, pp. 730–733 (2001)
5. Bo, S., Ma, Y., Zhu, C.: Image Segmentation by Nonparametric Color Clustering. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4489, pp. 898–901. Springer, Heidelberg (2007)

6. Haris, K., Efstratiadis, S., Maglaveras, N., Katsaggelos, A.: Hybrid image segmentation using watersheds and fast region merging. IEEE Trans. Image Proc. 7(12), 1684–1699 (1998)
7. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
8. Zadeh, L.: Fuzzy logic = computing with words. IEEE Trans. Fuzzy Systems 4(2), 103–111 (1996)
9. Tizhoosh, H.: Fuzzy ImageProcessing. Introduction in Theory and Practice. Springer, Berlin (1997)
10. Chi, Z., Yan, H., Pahm, T.: Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. In: Advances in Fuzzy Systems. Applications and Theory, 10. World Scientific Pub. Co. Inc (1996)
11. Sankowski, D., Strzecha, K., Jeżewski, S.: Digital image analysis in measurement of surface tension and wet ability angle. In: Int. Conf. Modern Problems of Telecommunications, Computer Science and Engineers Training, Lviv-Slavskie, Ukraine, pp. 129–130 (2000)
12. Sankowski, D., Senkara, J., Strzecha, K., Jeżewski, S.: Automatic investigation of surface phenomena in high temperature solid and liquid contacts. In: IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, pp. 1397–1400 (2001)
13. Adamson, A., Gast, A.: Physical Chemistry of Surfaces. Wiley-Interscience, USA (1997)
14. Dunn, J.: A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well Separated Clusters. J. Cyber. 3, 32–57 (1974)
15. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
16. Ridler, T., Calvard, S.: Picture thresholding using an iterative selection method. IEEE Trans. Syst. Man Cyb. 8, 630–632 (1978)
17. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman & Hall, London (1993)

IV Keynote 2

Challenges and Opportunities for Extracting Cardiovascular Risk Biomarkers from Imaging Data

I.A. Kakadiaris[1], E.G. Mendizabal-Ruiz[1], U. Kurkure[1], and M. Naghavi[2]

[1] Computational Biomedicine Lab, Departments of Computer Science,
Electrical and Computer Engineering, and Biomedical Engineering
University of Houston, Houston, TX
[2] Society for Heart Attack Prevention and Eradication (SHAPE), Houston, TX
http://www.cbl.uh.edu

Abstract. Complications attributed to cardiovascular diseases (CDV) are the leading cause of death worldwide. In the United States, sudden heart attack remains the number one cause of death and accounts for the majority of the $280 billion burden of cardiovascular diseases. In spite of the advancements in cardiovascular imaging techniques, the rate of deaths due to unpredicted heart attack remains high. Thus, novel computational tools are of critical need, in order to mine quantitative parameters from the imaging data for early detection of persons with a high likelihood of developing a heart attack in the near future (vulnerable patients). In this paper, we present our progress in the research of computational methods for the extraction of cardiovascular risk biomarkers from cardiovascular imaging data. In particular, we focus on the methods developed for the analysis of intravascular ultrasound (IVUS) data.

Keywords: vulnerable patients, intravascular ultrasound, vasa vasorum.

1 Introduction

The complications attributed to cardiovascular diseases (CDV) are the leading cause of death worldwide. For a significant percentage of patients, the first symptom of CVD is sudden death without previous warnings. In the United States, sudden heart attack remains the number one cause of death and accounts for the majority of the $280 billion burden of cardiovascular diseases. Cardiovascular specialists indicate that heart attacks are caused by inflammation of the coronary arteries and thrombotic complications of vulnerable plaques. The concept of vulnerable plaque has recently evolved into the definition of "vulnerable patient". *A vulnerable patient is defined as a person with more than a 10% likelihood of having a heart attack in the next 12 months.* Over 45 world leaders in cardiology have collectively introduced the field of vulnerable patient detection as defining the new era in preventive cardiology [1].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 227–235, 2009.

Detection of vulnerable patients is one of the most active areas of research in both the cardiology and biomedical imaging communities. While there exist many invasive and non-invasive medical imaging modalities for the study and diagnosis of CVD, until now none of them can completely identify a vulnerable plaque and accurately predict its further development. Therefore, the necessity of novel methods for extracting cardiovascular risk biomarkers from these image modalities is evident.

Cardiovascular risk biomarkers. The presence of calcified coronary plaque has a significant predictive value for coronary artery disease and is associated with with cardiovascular risk [2,3,4,5]. Furthermore, vasa vasorum (VV) neovascularization on the plaque has been identified as a common feature of inflammation [6] and has been defined as a plaque vulnerability index. Other cardiovascular risk biomarkers include calcification in the aorta and thoracic fat burden.

Our group is pioneering work in the development of computational methods for mining of information from different modalities of invasive and non-invasive cardiovascular imaging data. For the non-invasive image data analysis, using cardiac CT data, we have presented methods for the automatic detection of coronary artery regions [7,8], automatic detection of coronary calcium using a hierarchical supervised learning framework [9,10,11], automatic delineation of the inner thoracic region [12], and segmentation of thoracic aorta [13]. Additionally, the thoracic fat is detected using a relaxed version of multi-class, multi-feature fuzzy connectedness method [14].

Furthermore, our group has developed several computational methods for the analysis of intravascular ultrasound (IVUS) contrast imaging for the detection of VV in-vivo [15,16]. These methods include techniques for IVUS image stabilization and differential imaging techniques for the detection of those changes which occur in IVUS imagery due to the perfusion of an intravascularly-injected contrast agent into the plaque and vessel wall. However, one of the limitations of these methods is related to the use of the Cartesian B-mode representation of the IVUS signal. This is a disadvantage because the transformation of the ultrasound radio frequency (RF) signal data into this representation results in loss of potentially valuable information.

In this paper, we present ongoing work by our group in order to overcome these limitations and take full advantage of the information contained in the "raw" RF signal.

2 Methods

IVUS is currently the gold-standard technique for assessing the morphology of blood vessels and atherosclerotic plaques *in vivo*. The IVUS catheter consists of either a solid-state or a mechanically-rotated transducer which transmits a pulse and receives an acoustic signal at a discrete set of angles over each radial scan. Commonly, 240 to 360 such signals are obtained per (digital or mechanical) rotation. The envelopes of these signals are computed, log-compressed, and then geometrically transformed to obtain the familiar disc-shaped IVUS image.

IVUS has been combined with contrast-enhancing agents as blood tracers for the detection of blood perfusion due to VV [17,18,19,20]. The protocol for blood perfusion detection consists of injecting high-echogenic microbubbles of size similar to red blood cells into the blood flow while monitoring with IVUS. If these microbubbles are found beyond the lumen border, this could be an indication of microcirculation due to VV.

2.1 Contrast Agent Detection

Based on this protocol, we have investigated the feasibility of detecting microbubbles in IVUS data by acoustic characterization of the raw RF IVUS data using two approaches based on one-class cost-sensitive learning [21]. In the first approach, we built a model for the microbubbles from samples of microbubbles present in the lumen during the contrast agent injection. In the second approach, we detected the microbubbles as a change from baseline IVUS data consisting of samples of different tissues of the vessel extracted from frames before the injection.

For our models, we used those features based on frequency-domain spectral characterization that represent measures of high-frequency signal proposed by O'Malley *et al.* [16]. Specifically, these features are defined for a 3-D signal window of dimensions $r_0 \times \theta_0 \times t_0$ as follows:

$$F_\zeta = \sum_{i=1}^{\lceil r_0/2 \rceil} \sum_{j=1}^{\lceil \theta_0/2 \rceil} \sum_{k=1}^{\lceil t_0/2 \rceil} ijk\hat{W}(i,j,k) \tag{1}$$

$$F_\eta = \frac{F_\zeta}{\sum_{i=1}^{\lceil r_0/2 \rceil} \sum_{j=1}^{\lceil \theta_0/2 \rceil} \sum_{k=1}^{\lceil t_0/2 \rceil} \hat{W}(i,j,k)}, \tag{2}$$

where $\hat{W}$ indicates the magnitude of the Fourier spectrum of the windowed signal W. Each feature is computed on I_e and I_l in addition to I. Hence, each feature is a vector of three values. The samples are extracted by placing a 3-D fixed size window (r_0, θ_0, t_0) around each sample in the volume. These features are computed for this window and associated with the class contained by it. To improve the scaling of the feature space, each feature of the samples used for training is normalized to zero mean and unit variance. The normalization values are retained for use in testing and deployment. The parameters of the one-class SVM γ and ν are selected in such a way that good performance on the recognition of the class of importance and on the rejection of the negative class is obtained. However, since it is possible to have higher accuracy on the classification of negative samples than in the class of interest, we constrain the selection of parameters to provide an accuracy on the class of interest as close to 100% as possible. Therefore, the criteria for the selection of the best parameters is given by a weighted linear combination of the accuracy on the classification of both classes, $A = w_1 A_P + w_2 A_N$, where A stands for total accuracy, A_P and A_N

are the accuracies of detecting the class of interest and rejecting the negative class, respectively, and w_1 and $w_2 \in [0, 1]$ are the weights associated with the class of interest and negative class accuracy, respectively. This can be considered cost-sensitive learning for one-class classifiers.

2.2 Scattering Model-Based Analysis of IVUS

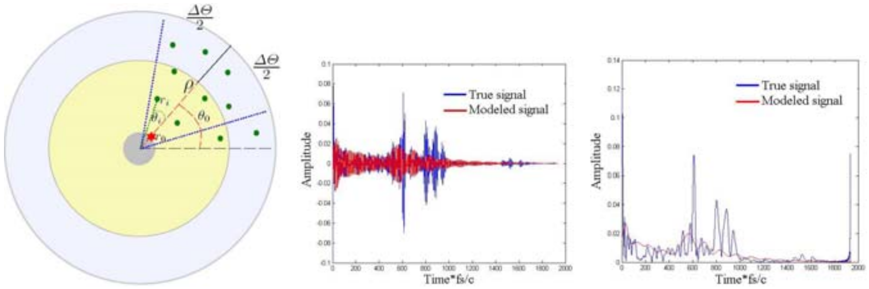
Currently, we are investigating the feasibility of using a physics-based scattering model of the IVUS RF signal for the analysis of the IVUS data. This model assumes that the IVUS signal can be obtained from a physical model based on the transmission and reflection of ultrasound waves that radially penetrate the arterial structure. Since the wavelength produced by IVUS transducers is very large in comparison to the dimension of the structures of the vessel, this model assumes that structures can be modeled as a finite set of point scatterers with an associated differential backscattering cross-section coefficient (DBC). In this model, the ultrasound beam interacts with scatterers along its radial direction along an angular window given by $\varDelta\varTheta = \sin^{-1}(1.22\frac{\lambda}{D})$ (Fig. 1(a)), where $\lambda = \frac{c}{f}$ is the wavelength, f is the transducer frequency and D is the transducer diameter. Assuming Born approximation scattering, we use the principle of superposition to represent the total scattered wave as a sum of reflections from individual point scatterers [22]. Then, using this model, the ultrasound reflected signal for each transducer's angular position $\varTheta_k$ at time t for a finite set of N scatterers with coordinates (r_i, θ_i) where $\theta_i \in \{\varTheta_k - \frac{\varDelta\varTheta}{2}, \varTheta_k + \frac{\varDelta\varTheta}{2}\}$ and DBC $\kappa(r_i, \theta_i)$ is given by:

$$\hat{S}(t, \varTheta_k) = \frac{1}{N} \sum_{i=1}^{N} \frac{\kappa(r_i, \theta_i) \exp(-\mu r_i)}{r_i} \exp\left(\frac{-(t - \frac{2r_i}{c})^2}{2\sigma^2} \right) \sin\left(\omega t - \frac{2r_i}{c} \right), \quad (3)$$

where μ is the attenuation coefficient, C defines the transducer constant parameters, and $\omega = 2\pi f$ is the angular velocity of the impulse function with width σ.

In order to be able to use this model, first it is necessary to recover its parameters. We accomplish this by solving an inverse problem on which we tune the parameters by the minimization of the difference between a ground truth signal and our modeled signal.

A significant difficulty is that the modeled signal depends on the position of the scatterers. We cannot treat the distribution of scatterers in a deterministic fashion: the scatterers' positions are the result of a spatial stochastic point process. Therefore, the minimization of the differences of the signals should be approached in a stochastic sense. We consider the optimal parameter values as functions of the scatterer locations. Then, for each angle k we generate ξ samplings of scatterer positions and minimize the sum of the errors between the real IVUS signal and each of the ξ modeled signals. Specifically, we solve the problem:



(a) (b) (c)

Fig. 1. (a) Scatterers interacting with the ultrasound beam on IVUS. (b) Raw real and modeled IVUS signals for a single angle. (c) Positive envelope of real and modeled IVUS signals for a single angle.

$$\min_{\sigma_k,\kappa_k^l,\kappa_k^w} \frac{1}{2}\sum_t \sum_{i=1}^{\xi} (E(t,\Theta_k) - \hat{E}_i(t,\Theta_k,\sigma,\kappa^l,\kappa^w))^2 \ , \tag{4}$$

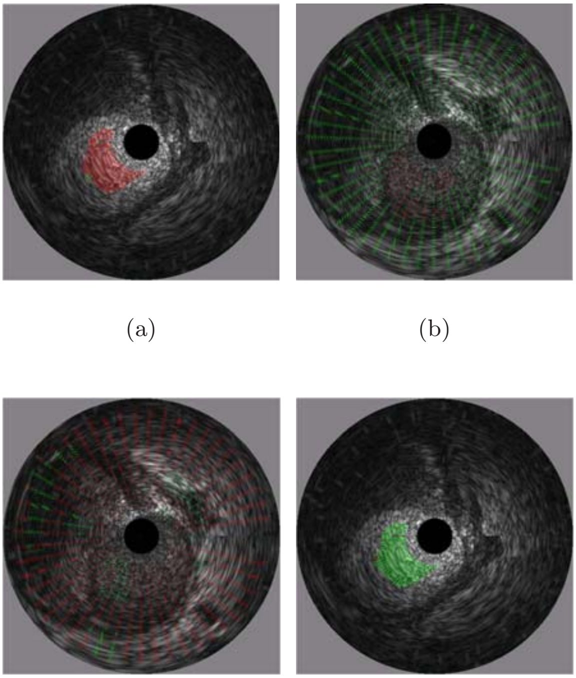
where $E(t,\Theta_k)$ and $\hat{E}_i(t,\Theta_k,\sigma,\kappa^l,\kappa^w)$ are the positive envelopes for the real and the modeled signals, respectively (Figs. 1(b) and 1(c)).

We have applied this model to the segmentation of the luminal border using the IVUS RF data [23]. For this, we consider that the radial position ρ_k of the lumen border for each angle Θ_k can be recovered by solving an inverse problem as well. We use the parameters computed on the calibration and we find ρ_k by the minimization of the sum of differences between the real IVUS signal $S(t,\Theta_k)$ and the signals computed with our model $\hat{S}_i(t,\Theta_k,\rho_k)$ for each sampling ξ. Specifically, we solve:

$$\min_{\rho_k} \frac{1}{2}\sum_t \sum_{i=1}^{\xi} (E(t,\Theta_k) - \hat{E}_i(t,\Theta_k,\rho_k))^2 \ . \tag{5}$$

3 Results

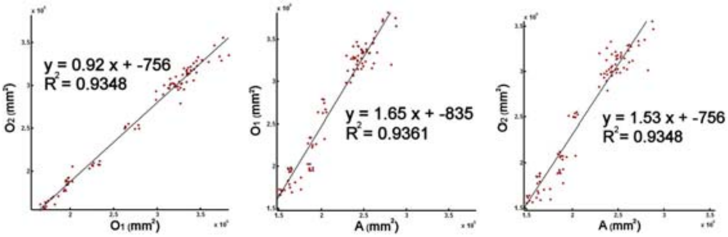
Regarding the detection of contrast agent, for the first approach, we obtained an average accuracy of 99.17% on the detection of microbubbles on lumen and 91.67% on the classification of pre-injection frames as having no microbubbles, with an average percentage of support vectors less than 1% of the total training samples. For the second approach, we obtained an average accuracy of 89.65% on the detection of baseline IVUS data and 96.78% on the classification of microbubbles as change, with an average percentage of support vectors less than 10% of the total number of samples used for training. Figure 2 depicts examples of the classification results on frames before injection and during injection



(a) (b)

(c) (d)

Fig. 2. Classification results in (a,d) a frame with microbubbles in the lumen and (b,c) an IVUS frame before injection. For the first approach (a) and (b), the red color indicates the pixels classified as microbubbles and the green color those classified as non-microbubbles. For the second approach (c) and (d), the red color indicates the pixels classified as baseline IVUS and the green color those classified as an anomaly.



(a) (b) (c)

Fig. 3. Linear regression plot for (a) O_1 vs O_2, (b) A vs. O_1 and (c) A vs. O_2. Each point corresponds to one of the 90 segmented frames.

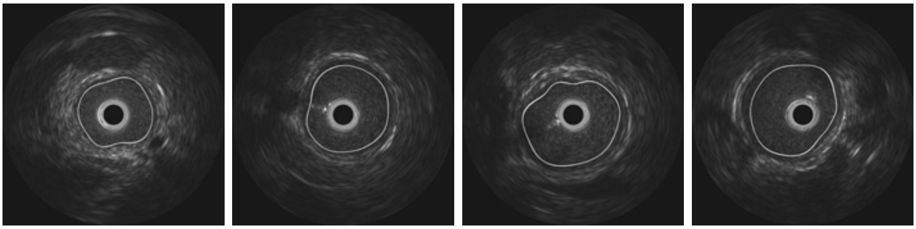


Fig. 4. Examples of segmentation results

for both approaches. One of the advantages of this methodology is that by using one-class learning, we did not need to provide "background" samples for building the models. In our case this was important because, although samples for microbubbles in lumen can be easily acquired by manual annotations from an expert, the background can consist of a wide variety of other imaged tissues. Thus, obtaining samples for the other tissues may be difficult and labor-intensive.

We test our RF-based segmentation method on 90 frames from 40MHz sequences and the results were evaluated by comparing the agreement between areas corresponding to lumen on each frame by our method (A) with manual segmentations from two expert observers (O_1 and O_2). The resulting mean Dice similarity coefficient was $s = 90.27$. In addition, we performed linear regression. The coefficient of determination (R^2, where R is the linear correlation) for area differences between O_1 and O_2 (O_1,O_2) was R^2=0.98, and R^2=0.93 and R^2=0.93 for (A,O_1) and (A,O_2), respectively. Figure 3 depicts the results of this analysis and Fig. 4 depicts examples of the segmentation results.

4 Conclusions

We have presented methods for the analysis of IVUS data based on the RF signal. Future developments in VV detection methods will consist of using the scattering model to extract novel features to be used in combination with machine learning techniques for the detection of contrast agent within the plaque. The techniques presented in this paper may contribute significantly in the detection of neovascularization within atherosclerotic plaques. However, since VV is not the only cardiovascular risk biomarker, the combination of data from IVUS and other imaging modalities is necessary in order to provide an effective way to detect vulnerable patients. The expected impact of our work stems from the fact that sudden heart attack remains the number one cause of death in the US, and unpredicted heart attacks account for the majority of the $280 billion burden of cardiovascular diseases.

References

1. Naghavi, M., Libby, P., Falk, E., Casscells, S., Litovsky, S., Rumberger, J., Badimon, J., Stefanadis, C., Moreno, P., Pasterkamp, G., Fayad, Z., Stone, P., Waxman, S., Raggi, P., Madjid, M., Zarrabi, A., Burke, A., Yuan, C., Fitzgerald, P., Siscovick, D., de Korte, C., Aikawa, M., Airaksinen, K., Assmann, G., Becker, C., Chesebro, J., Farb, A., Galis, Z., Jackson, C., Jang, I., Koenig, W., Lodder, R., March, K., Demirovic, J., Navab, M., Priori, S., Rekhter, M., Bahr, R., Grundy, S., Mehran, R., Colombo, A., Boerwinkle, E., Ballantyne, C., Insull, J., Schwartz, W.R., Vogel, R., Serruys, P., Hansson, G., Faxon, D., Kaul, S., Drexler, H., Greenland, P., Muller, J., Virmani, R., Ridker, P., Zipes, D., Shah, P., Willerson, J.: From vulnerable plaque to vulnerable patient: A call for new definitions and risk assessment strategies: Part I. Circulation 108(14), 1664–1672 (2003)
2. Greenland, P., LaBree, L., Azen, S., Doherty, T., Detrano, R.: Coronary artery calcium score combined with Framingham score for risk prediction in asymptomatic individuals. Journal of the American Medical Association 291(2), 210–215 (2004)
3. Shaw, L., Raggi, P., Schisterman, E., Berman, D., Callister, T.: Prognostic value of cardiac risk factors and coronary artery calcium screening for all-cause mortality. Radiology 228(3), 826–833 (2003)
4. Taylor, A., Bindeman, J., Feuerstein, I., Cao, F., Brazaitis, M., O'Malley, P.: Coronary calcium independently predicts incident premature coronary heart disease over measured cardiovascular risk factors: Mean three-year outcomes in the Prospective Army Coronary Calcium (PACC) project. Journal of the American College of Cardiology 46(5), 807–814 (2005)
5. Wong, N.D., Budoff, M.J., Pio, J., Detrano, R.C.: Coronary calcium and cardiovascular event risk: Evaluation by age- and sex-specific quartiles. American Heart Journal 143(3), 456–459 (2002)
6. Gossl, M., Malyar, N., Rosol, M., Beighley, P., Ritman, E.: Impact of coronary vasa vasorum functional structure on coronary vessel wall perfusion distribution. American Journal of Physiology - Heart and Circulatory Physiology 285(5), H2019–H2026 (2003)
7. Brunner, G., Chittajallu, D., Kurkure, U., Kakadiaris, I.: Toward the automatic detection of coronary artery regions in non-contrast Computed Tomography data. International Journal of Cardiovascular Imaging (in press, 2009)
8. Brunner, G., Chittajallu, D., Kurkure, U., Kakadiaris, I.: A heart-centered coordinate system for the detection of coronary artery zones in non-contrast Computed Tomography data. In: Proc. Medical Image Computing and Computer-Assisted Intervention Workshop on Computer Vision for Intravascular and Intracardiac Imaging, New York, NY, September 10 (2008)
9. Kurkure, U., Chittajallu, D., Brunner, G., Yalamanchili, R., Kakadiaris, I.: A supervised classification-based method for coronary calcium detection in non-contrast CT. International Journal of Cardiovascular Imaging (in press, 2009)
10. Kurkure, U., Chittajallu, D., Brunner, G., Yalamanchili, R., Kakadiaris, I.: Detection of coronary calcifications using supervised hierarchical classification. In: Proc. Medical Image Computing and Computer-Assisted Intervention Workshop on Computer Vision for Intravascular and Intracardiac Imaging, New York, NY, September 10 (2008)
11. Brunner, G., Kurkure, U., Chittajallu, D., Yalamanchili, R., Kakadiaris, I.: Toward unsupervised classification of calcified arterial lesions. In: Proc. 11th International Conference on Medical Image Computing and Computer Assisted Intervention, New York, NY, September 6-9, pp. 144–152 (2008)

12. Chittajallu, D., Balanca, P., Kakadiaris, I.: Automatic delineation of the inner thoracic region in non-contrast CT data. In: Proc. 31st International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, September 2-6 (2009)
13. Kurkure, U., Avila-Montes, O., Kakadiaris, I.: Automated segmentation of thoracic aorta in non-contrast CT images. In: Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Paris, France, May 14-17 (2008)
14. Pednekar, A., Kurkure, U., Kakadiaris, I., Muthupillai, R., Flamm, S.: Left ventricular segmentation in MR using hierarchical multi-class multi-feature fuzzy connectedness. In: Proc. 7th International Conference on Medical Image Computing and Computer Assisted Intervention, Rennes, Saint-Malo, France, September 26-30, pp. 402–410 (2004)
15. O'Malley, S., Vavuranakis, M., Naghavi, M., Kakadiaris, I.: Intravascular ultrasound-based imaging of vasa vasorum for the detection of vulnerable atherosclerotic plaque. In: Proc. Medical Image Computing and Computer-Assisted Intervention, Palm Springs, CA, October, pp. 343–351 (2005)
16. O'Malley, S., Naghavi, M., Kakadiaris, I.: One-class acoustic characterization applied to blood detection in IVUS. In: Proc. 10th International Conference on Medical Image Computing and Computer Assisted Intervention, Brisbane, Australia, October 29 - November 2, pp. 202–209 (2007)
17. Papaioannou, T., Vavuranakis, M., Androulakis, A., Lazaros, G., Kakadiaris, I., Vlaseros, I., Naghavi, M., Kallikazaros, I., Stefanadis, C.: In-vivo imaging of carotid plaque neoangiogenesis with contrast-enhanced harmonic ultrasound. International Journal of Cardiology 134(3), 110–112 (2009)
18. Vavuranakis, M., Kakadiaris, I., O'Malley, S., Papaioannou, T., Sanidas, E., Naghavi, M., Carlier, S., Tousoulis, D., Stefanadis, C.: A new method for assessment of plaque vulnerability based on vasa vasorum imaging, by using contrast-enhanced intravascular ultrasound and automated differential image analysis. International Journal of Cardiology 130, 23–29 (2008)
19. Vavuranakis, M., Kakadiaris, I., O'Malley, S., Papaioannou, T., Carlier, S., Naghavi, M., Stefanadis, C.: Contrast-enhanced intravascular ultrasound: Combining morphology with activity-based assessment of plaque vulnerability. Expert Review of Cardiovascular Therapy 5, 915–917 (2007)
20. Vavuranakis, M., Papaioannou, T., Kakadiaris, I., O'Malley, S., Naghavi, M., Filis, K., Sanidas, E., Papalois, A., Stamatopoulos, I., Stefanadis, C.: Detection of perivascular blood flow in vivo by contrast-enhanced intracoronary ultrasonography and image analysis: An animal study. Clinical and Experimental Pharmacology and Physiology 34(12), 1319–1323 (2007)
21. Mendizabal-Ruiz, E., Kakadiaris, I.: One-class acoustic characterization applied to contrast agent detection in IVUS. In: Proc. Medical Image Computing and Computer-Assisted Intervention Workshop on Computer Vision for Intravascular and Intracardiac Imaging, New York, NY, September 10 (2008)
22. Fontaine, I., Bertrand, M., Cloutier, G.: A system-based approach to modeling the ultrasound signal backscattered by red blood cells. Biophysical Journal 77(5), 2387–2399 (1999)
23. Mendizabal-Ruiz, E., Biros, G., Kakadiaris, I.: An inverse scattering algorithm for the segmentation of the luminal border on intravascular ultrasound data. In: Proc. 12th International Conference on Medical Image Computing and Computer Assisted Intervention, London, UK, September 20-24 (2009)

V Geometric Image Processing and Analysis

A New Unsupervised Learning for Clustering Using Geometric Associative Memories*

Benjamín Cruz, Ricardo Barrón, and Humberto Sossa

Center for Computing Research - National Polytechnic Institute,
México City 07738, México
benji@helgrind.net, rbarron@cic.ipn.mx, hsossa@cic.ipn.mx
http://www.cic.ipn.mx

Abstract. Associative memories (AMs) have been extensively used during the last 40 years for pattern classification and pattern restoration. A new type of AMs have been developed recently, the so-called Geometric Associative Memories (GAMs), these make use of Conformal Geometric Algebra (CGA) operators and operations for their working. GAM's, at the beginning, were developed for supervised classification, getting good results. In this work an algorithm for unsupervised learning with GAMs will be introduced. This new idea is a variation of the k-means algorithm that takes into account the patterns of the a specific cluster and the patterns of another clusters to generate a separation surface. Numerical examples are presented to show the functioning of the new algorithm.

1 Introduction

Associative Memories (AMs) have been extensively used for many years in pattern recognition problems. An AM can be seen as an input-output system, see ([1]). When an input pattern x is presented to an AM, it must to respond with the corresponding output pattern y.

$$x \rightarrow M \rightarrow y \ . \tag{1}$$

The associative memories models developed until now can be categorized in three groups, those based on traditional vector algebra operations, those based on mathematical morphology operations, and those based on Geometric Algebra paradigm. The third group make use of Geometric Algebra [1] for their operations an operators, the so-called *Geometric Associative Memories* [2] (GAMs) are an example of memories that falls into this group.

The goal of GAMs is the classification of a pattern as belonging to a specific class if and only if the pattern is inside of the support region (hyper-sphere) of that class. Originally, GAMs were developed to function in a supervised form. In

* This work has been supported by the National Polytechnic Institute of Mexico (SIP-IPN), under grants 20090620 and 20091421, by the Mexican Science and Technology National Council (CONACyT) under grants 46805 and 182938, and the European Comission and CONACyT under grant FONCICYT 93829.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 239–246, 2009.

this work, a new unsupervised learning algorithm for GAMs will be developed, it will be based on the well-known k-means [11] algorithm idea, but it will be use operators and operations of CGA for the building of the respective cluster.

2 Basics on Geometric Algebra

Geometric Algebras (GA's) also known as Clifford Algebras were introduced by William K. Clifford in 1878. He joined the works of Grassmann with the quaternion of Hamilton into a new mathematical model [1]. GA is a free coordinate geometric schema [5]. In GA, the geometric objects and the operators over these objects are treated in a single algebra [3]. A special characteristic of GA is its geometric intuition. Another important feature is that, the expressions in GA usually have low symbolic complexity [7].

The Conformal Geometric Algebra (CGA) is a coordinate-free theory. In CGA, spheres and circles are both algebraic objects with a geometric meaning. In CGA, points, spheres, and planes are easily represented as *multivectors*. A multivector is the outer product of various vectors. CGA provides a great variety of basic geometric entities to compute with [7]. In CGA the inner product is used for the computation of angles and distances.

For notations purposes, Euclidean vectors will be noted by lowercase italic letters (p,q,s), with exception of the letters:i,j,k,l,m,n; these will be used to refer to indexes. The corresponding conformal points will be noted by uppercase italic letters (C,P,S). A Euclidean matrix will be noted by a bold capital letter (**M,H**). To denote that an element belongs to an object (vector), a sub-script will be used. To refer that an object belongs to a set of objects of the same type, a superscript will be used. For example, let S be a sphere, then S_k is the k-th component of it, and S^k is the k-th sphere of a set of spheres. To denote scalars Greek letters will be used $(\gamma, \varepsilon, \delta)$.

Let $p \in \mathbb{R}^n$ be an Euclidean point, it can be transformed to a CGA representation as:

$$P = p + \frac{1}{2}(p)^2 e_\infty + e_0 \ , \tag{2}$$

where e_0 is the Euclidean origin and e_∞ is the point at infinity such that $e_0^2 = e_\infty^2 = 0$ and $e_0 \cdot e_\infty = -1$. Let P and Q two conformal points, the distance between them is found by means of the inner product [6] as follows:

$$P \cdot Q = p \cdot q - \frac{1}{2}(p)^2 - \frac{1}{2}(q)^2 = -\frac{1}{2}(p-q)^2 \Longleftrightarrow (p-q)^2 = -2(P \cdot Q) \ . \tag{3}$$

In the same way, a sphere takes the following representation [8]:

$$S = C - \frac{1}{2}(\gamma)^2 e_\infty = c + \frac{1}{2}\big((c)^2 - (\gamma)^2\big)e_\infty + e_0 \ , \tag{4}$$

where C is the center point of the sphere in conformal notation as defined in (2), γ is the radius of the sphere and c is the Euclidean point of C.

A distance measure between one conformal point P and a sphere S can be defined with the help of the inner product [10], as follows:

$$P \cdot S = p \cdot s - \frac{1}{2}\left((s)^2 - \frac{1}{2}(\gamma)^2\right) - \frac{1}{2}(p)^2 = \frac{1}{2}\left((\gamma)^2 - (s-p)^2\right) \ , \tag{5}$$

or in simplified form:

$$2(P \cdot S) = (\gamma)^2 - (s-p)^2 \ . \tag{6}$$

Based on (6):

1. If $(P \cdot S > 0)$ then P is inside of S.
2. If $(P \cdot S < 0)$ then P is outside of S.
3. If $(P \cdot S = 0)$ then P is on S.

Therefore, in pattern classification if a CGA spherical neighborhood is used as support region, with the help of the inner product it is possible to know when a pattern is inside or outside of the region. In the same way, the distance between two points is easily computed with their inner product.

3 Geometric Associative Memories for Clustering

GAMs were developed, in principle, as a supervised classification model [2]. The training phase of that model is done by finding an optimal sphere with quadratic programming. In the classification phase an inner product between the unclassified pattern and the GAM must be applied. Then a minimum function is used to obtain an index class. GAMs can perfectly operate when the classes are spherically separable [2].

A GAM is precisely a matrix whose components are spheres, it can be seen in (7), where m is the total number of classes. It uses spherical neighborhoods as decision regions.

$$\mathbf{M} = \begin{bmatrix} S^1 \\ S^2 \\ \vdots \\ S^m \end{bmatrix} \ . \tag{7}$$

Often, a clear distinction is made between learning problems than are supervised (classification) o unsupervised (clustering), the first one involving only labeled data while the latter involving only unlabeled data [4]. However, clustering is a more difficult and challenging problem than classification [9]. The goal of data clustering is to find the *natural* grouping in a set of patterns, points, or objects without any knowledge of class label. In other words, it consists in to develop an automatic algorithm that will discover the natural grouping in the unlabeled data.

The K-means [11] is one of the simplest unsupervised learning algorithms that solve the clustering problem. The main idea is to define k random centroids, one for each cluster. Then, to take each point of the data set and associate it to the nearest centroid using a distance measure. The new centroids must be recalculated as the centers of the clusters found. Then, with the new centers, the second step must be done, a loop has been generated. The loop is repeated until no more changes of centers are done.

The proposed algorithm is based on the previous ideas, but the Conformal Geometric Algebra paradigm will be used. Different to the k-means algorithm the clusters in this method will be found by separating the points of the associated centroid of the others points (the main advantage is that the new centroids are found automatically) solving an optimization problem, it is described in [2].

Given two sets of points $\left\{p^i\right\}_{i=1}^{l}$ and $\left\{p^j\right\}_{j=l+1}^{m}$, the idea is to find an optimal sphere S with the least square error, such that $\{p^i\}$ are inside of S and $\{p^j\}$ are outside of it. In other words, to solve:

$$\min_{S} = \sum_{i=1}^{m}(P^i \cdot S) \ , \tag{8}$$

subject to (9) for points inside of the sphere and (10) for points outside of it.

$$P^i \cdot S \geq 0, i = 1, \ldots, l \ . \tag{9}$$

$$P^j \cdot S < 0, j = l + 1, \ldots, l \ . \tag{10}$$

That method takes into account the points inside and the points outside of the classification sphere for a better performance. Spheres function like attractors for the inside points and like contractors for the outside points, this gives a optimal separation surface. Another improvement is the computing of the distances among the centroids and the sets of points. This procedure is done with a GAM. The GAM is composed with the corresponding centroids. At the first iteration the centroids are randomly generated, at the following iterations the centroids are computing automatically when the corresponding convex hulls are built.

3.1 New Clustering Algorithm

Given a set of points $\left\{p^i\right\}_{i=1}^{m}$ in $\mathbb{R}^n$, and let k the number of clusters:

1. Change p^i to its conformal representation P^i, by using expr. (2), for $i = 1, \ldots, m$.
2. Set k random centroids, C^j where $j = 1, \ldots, k$.
3. Set $\mathbf{M} = [C^1, \ldots, C^k]'$
4. Generate k clusters by associating every P^i with the nearest centroids. It can be done with $arg \min(M \cdot P^i)$ for $i = 1, \ldots, m$.
5. Find k spherical surfaces S^j for each cluster by using eq. (8).
6. New values of k centroids are the centers of each sphere.
7. Repeat step 3 until the values of C^j do not change.

As can be observed, in this algorithm the cluster computing is done by a single inner product and a minimum function, unlike the traditional k-means algorithm where a nearest neighbor algorithm to find the nearest centroid is used. Regarding computational complexity of both algorithms, it is known that the k-means algorithm can be solved in time $\Theta\left(m^{nk+1}\log m\right)$ [12], where m is the number of points to be clustered, n is the dimension of that points, and k is the number of clusters.

In the case of the proposal, most of the time is spent on generating cluster associations among points and clusters (step 4). One such operation costs $\Theta(k)$. The new clusters generation costs $\Theta(kmn)$, so its overall complexity is $\Theta(k^2mn)$. For a fixed number of iterations I, the overall complexity is therefore $\Theta(Ik^2mn^2)$. Thus, our proposal has a linear complexity into the Conformal Geometric Algebra framework.

Another improvement of our algorithm is that at the end, in some situations, some points can be outside of all the spheres which can be both and advantage or disadvantage, depending on the characteristics of the problem.

Sometimes, that points can be noise. And they can be omitted without lossing essential information about the nature of the problem. But, they can be essential for most clustering problems.

4 A Representative Example

Due space limitations, one numerical example will be shown. Let the set of points in $\mathbb{R}^2$ shown in Figure 1 to be clustered. In this image, as it can be observed, four cumulus of points are visible at first appearance.

When the algorithm of the section 3.1 is applied, the graphs of the Figures 2, 3, and 4 are obtained. The solution in each case, as in most clustering problems, depends, mainly, of the first random centroids. Best solution is shown in Figure 4.

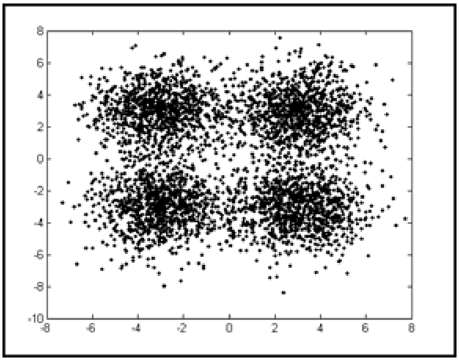


Fig. 1. Set of points in $\mathbb{R}^2$ to be clustered

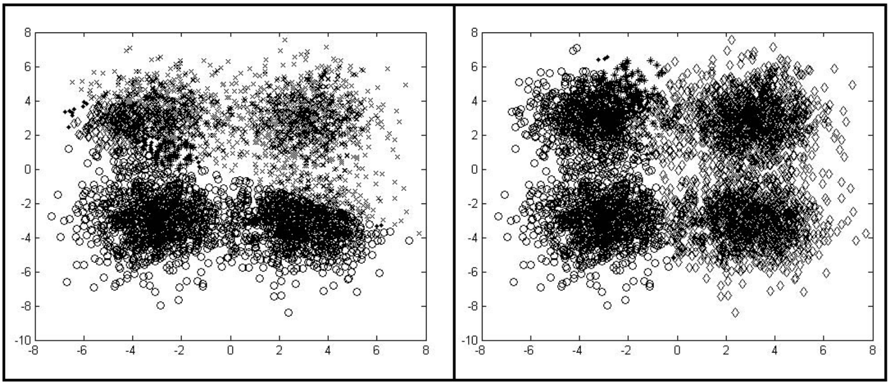


Fig. 2. Different solutions that solve the clustering problem of Fig. 1

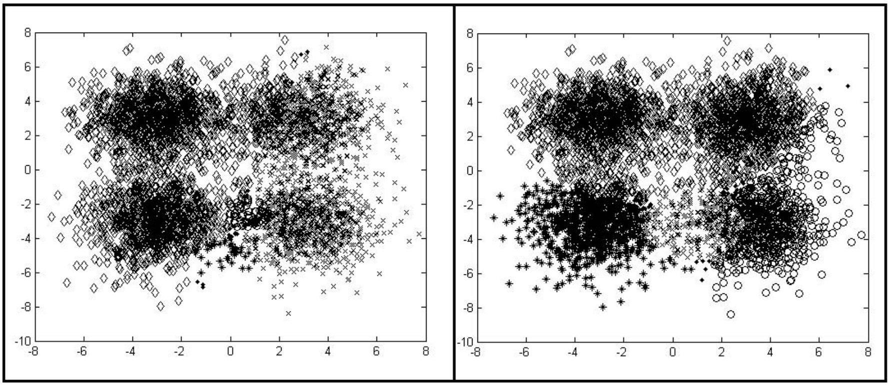


Fig. 3. Another solutions that solve the clustering problem of Fig. 1

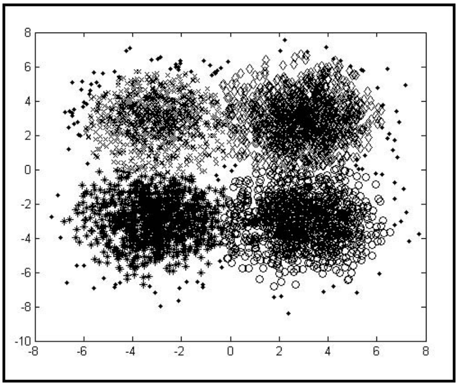


Fig. 4. Better solution that solves the clustering problem of Fig. 1

In most of the graphs presented, some black points appeared, these points are outside of all the spheres, furthermore these pointds do not belong to a specific class. This is due the type of separation surface that is being used. A better approximation can be obtained with the use of other irregular shapes unlike the spheres or squares. But, depending on the type of problem this could be an advantage

5 Conclusions and Ongoing Work

An Associative Memory (AM) is an input-output device used in pattern recognition and pattern classification. Many AM models make use of traditional algebra or mathematical morphology for their working. Essentially, AMs were developed in a principle for supervised classification or pattern recognition. Recently a new kind of AM was developed, the so-called Geometric Associative Memory (GAM) are based on Conformal Geometric Algebra (CGA) paradigm for their operations and operators.

GAMs have been developed for supervised pattern classification getting good results. In this work a new algorithm for GAMs was introduced, this new algorithm can solve the clustering problem, it is based on the well-known k-means algorithm, but it works into the CGA framework.

New clusters are generated using an inner product between the GAM itself and one unclassified point, unlike the traditional k-means algorithm where all the distances among the current centroids and each point must be applied. It reduces the complexity of the algorithm.

Unlike traditional k-means algorithm. In the proposed algorithm some initial points can not be considered as belonging to a specific class. In some type of problems this can be an advantage because these points could be spurious points or noise. One numerical example was presented to test the algorithm. Most of the cases, best solution depends, mainly, of the first random centroids.

Nowadays, we are also interested to test our method in more realistic situations and in comparison (in computing time and performance) between the proposed model and other clustering models. We are working too in GAMs that work with separation surfaces other than spheres; like ellipses, squares or another irregular shape.

References

1. Clifford, W.: Applications of Grassmann's Extensive Algebra. Am. J. of Math. 1(4), 350–358 (1878)
2. Cruz, B., Barrón, R., Sossa, H.: Geometric Associative Memories and their Applications to Pattern Classification. In: Bayro-Corrochano, E., Sheuermann, G. (eds.) Geometric Algebra Computing for Computing Science and Engineering. Springer, London (2009) (to be published)
3. Dorst, L., Fontijne, D.: 3D Euclidean Geometry Through Conformal Geometric Algebra (a GAViewer Tutorial). University of Amsterdam Geometric Algebra Website (2005), http://www.science.uva.nl/ga/tutorials/CGA/

4. Duda, R., HArt, P., Stork, D.: Pattern Classification. John Wiley & Sons, New York (2001)
5. Hestenes, D., Sobczyk, G.: Clifford Algebra to Geometric Calculus. Kluwer/Springer (1984)
6. Hestenes, D., Li, H., Rockwood, A.: New Algebraic Tools for Classical Geometry. In: Sommer, G. (ed.) Geometric Computing with Clifford Algebras, vol. 40, pp. 3–23. Springer, Heildeberg (2001)
7. Hildebrand, D.: Geometric Computing in Computer Graphics Using Conformal Geometric Algebra. Tutorial, TU Darmstadt, Germany. Interact. Graph. Syst. Group (2005)
8. Hitzer, E.: Euclidean Geometric Objects in the Clifford Geometric Algebra of Origin, 3-Space. Infinity. Bull. of the Belgian Math. Soc. 11(5), 653–662 (2004)
9. Jain, A.: Data Clustering: 50 Years Beyond K-means. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 3–4. Springer, Heidelberg (2008)
10. Li, H., Hestenes, D., Rockwood, A.: Generalized Homogeneous Coordinates for Computational Geometry. In: Sommer, G. (ed.) Geometric Computing with Clifford Algebras, vol. 40, pp. 27–52. Springer, Heildeberg (2001)
11. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. of 5-th Berkeley Symp. on Math. Stats. and Prob., vol. 1, pp. 281–297. Unversity of California Press, Berkeley (1967)
12. Inaba, M., Katoh, N., Imai, H.: Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In: Proceedings of 10th ACM Symposium on Computational Geometry, pp. 332–339 (2004)

Airway Tree Segmentation from CT Scans Using Gradient-Guided 3D Region Growing

Anna Fabijańska, Marcin Janaszewski, Michał Postolski, and Laurent Babout

Computer Engineering Department, Technical University of Lodz
18/22 Stefanowskiego Str., 90-924, Lodz, Poland
{an_fab,janasz,mpostol,lbabout}@kis.p.lodz.pl

Abstract. In this paper a new approach to CT based investigation of pulmonary airways is introduced. Especially a new - fully automated algorithm for airway tree segmentation is proposed. The algorithm is based on 3D seeded region growing. However in opposite to traditional approaches region growing is applied twice: firstly – for detecting main bronchi, secondly – for localizing low order parts of the airway tree. The growth of distal parts of the airway tree is driven by a map constructed on the basis of morphological gradient.

Keywords: CT, airway tree, image segmentation, 3D region growing.

1 Introduction

CT chest scans are commonly used for investigation of pulmonary disorders, especially chronic obstructive pulmonary disease (COPD) which is a common name for pathological changes characterized by airflow limitation due to different combinations of airway disease and emphysema [1][2]. The thickness of an airway wall and diameter of an airway lumen can provide important information about many pulmonary diseases therefore identification of airways in CT scans is an important step for many clinical applications and for physiological studies [2].

Traditionally, analysis of chest CT scans was preformed by radiologists who recognized areas of abnormal airway properties in consecutive slices of the examined scan. However, analyzing about 400 slices covering chest area is too cumbersome for everyday clinical use. Moreover analysis performed manually resulted in subjective and qualitative estimation of airway abnormalities without accurate quantification of pathological changes.

The main goal of recent clinical applications is to provide a useful tool for characterizing airway data. The major challenge of such applications is airway tree segmentation from CT scans. It is very difficult due to inhomogenity of a bronchial lumen, adjacency of the blood vessels, and changes of grey levels along airway walls.

In this paper problem of airway tree reconstruction from volumetric CT data is considered. Especially, segmentation using modified, 3D region growing method is regarded. In the proposed approach the growth of an airway tree is guided and constrained by morphological gradient.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 247–254, 2009.

2 Airway Tree

An airway tree is a part of respiratory tract that conducts air into the lungs. It starts with trachea which splits into two main bronchi: the left and the right one. The main bronchi subdivide into two segmental bronchi. These in turn split into twigs called bronchioles. Divisions of the airways into the smaller ones define orders of bronchi.

An illustrative image of the airway tree is presented in Figure 1. Consecutive numbers denote trachea (1), left (2a) and right main bronchus (2b), bronchi (3) and bronchioles (4). All branches of the airway tree are composed from airway lumen surrounded by high-density vascular airway wall. An airway lumen is filled with air.

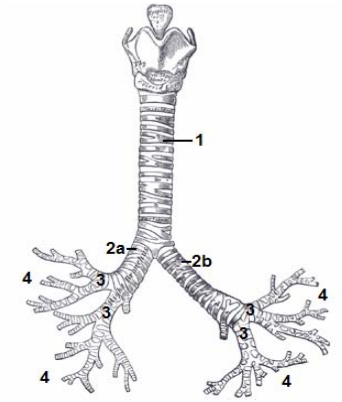


Fig. 1. 3D view of an airway tree; 1 – trachea; 2a – right main bronchus; 2b – left main bronchus; 3 – bronchi; 4 – bronchioles

3 Related Works

Different approaches to airway tree segmentation have been reported in the literature so far. However, in general they can be qualified as 3D approaches or 3D/2D (hybrid) approaches.

Three-dimensional approaches to airway tree segmentation act on a 3D volumetric image build up from series of planar CT images combined into a stack. Mostly, they involve seeded 3D region-growing supported by 3D propagation procedures such as combination of axial and radial propagation potentials [4] or non linear filters scanning for short airway sections [5] and connecting branch segments sharing common walls [6]. Up to 5 orders of bronchi could be detected using this method.

Hybrid techniques for airway tree reconstruction combine planar analysis with 3D segmentation. Usually, conventional 3D region-growing is firstly used to identify large airways. Secondly 2D analysis of consecutive CT slices is performed. It aims to define potential localization of candidate airways in planar images. Algorithms of candidate airways detection involve various techniques. The most important ones are:

- **Rule-based techniques** which utilize anatomical knowledge about airways and blood vessels. Candidate airways are recognized on individual slices based on *a priori* information about airways and vessels [7][8]. These techniques often suffer from large numbers of falsely detected airways.

- **Gray-level morphological techniques** which use grayscale morphological reconstruction to identify local extremes in the image. Candidate for airways and vessels on consecutive CT slices are considered as valleys and peaks in the grayscale profile of the current slice [9][10].
- **Wave front propagation techniques** which start from already detected airways and propagate waves in 2D plane to detect walls of the bronchi [11][12].
- **Template matching techniques** that search consecutive slices for oval dark rings surrounding brighter areas (airways) [12] or dark solid oval areas (adjacent blood vessels). Templates may be predefined *a priori* [13] or set adaptively [12].

Few 2D approaches to airway tree investigation were also reported. In these methods only planar analysis of consecutive CT slices is carried out [2]. However, 2D approaches are in minority because of their poor efficiency and low accuracy.

Having in mind classification presented above introduced method can be considered as a tree dimensional one. It extracts up to 8-9 generations of bronchi while avoiding leakages and falsely detected branches.

4 Input Data

3D volumetric CT chest scans of several patients were examined. They were obtained from GE LightSpeed VCT Scanner. The average number of transverse slices per each examination was 450 with the slice thickness equal to 0.625 mm. The slices were provided with 16-bit resolution and stored as signed 16-bit monochromatic images of the resolution 512x512 pixels. Individual slices were stacked into a 3D space representing volumetric data set. Exemplary CT slice with anatomical areas marked is presented in Figure 2.

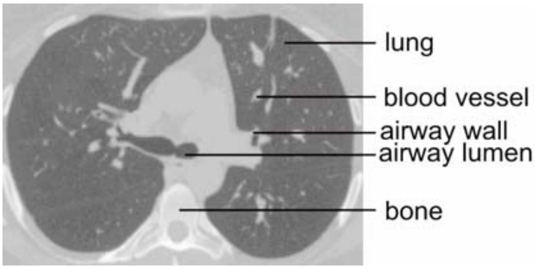


Fig. 2. Exemplary CT slice at the carina level with important anatomical areas marked

In analyzed images gray levels measured in Hunfield units (HU) ranged from -1024 to around 700. Grey levels which represent airway lumen and lung tissues filled with air were over -1024 HU but less than -850 HU. Airway walls, blood vessels and other high-density vascular tissues were areas of the intensities within the range -300:50 HU. Intensities over 300 HU matched hard tissues (bones).

5 Airway Tree Segmentation

5.1 3D Seeded Region Growing – The Main Idea

3D seeded region growing is a simple and convenient algorithm for image segmentation. In case of CT chest scans, the method starts from the seed point in the centre of trachea and builds a set of voxels by iteratively joining new similar pixels. In this way the whole airway tree should be segmented. This idea is presented on Figure 3a. However, as it was mentioned many times in the trade literature 3D region growing is not sufficient for segmentation of complete airway tree from CT scans due to its proneness to leakages.

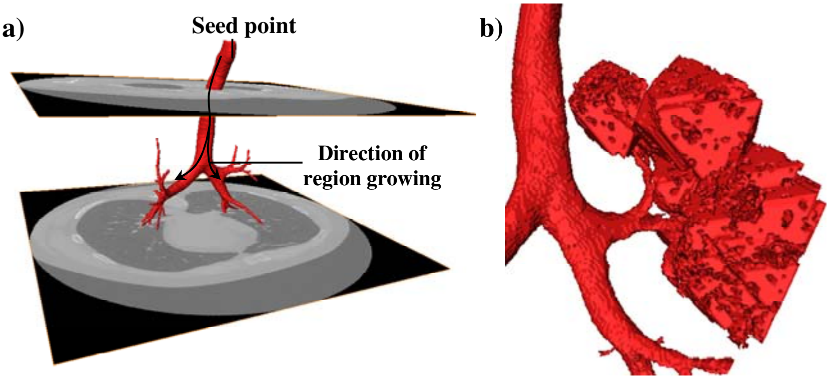


Fig. 3. Airway tree segmentation using seeded 3D region growing algorithm; a) the main idea; b) algorithm leaking into the lung parenchyma

On CT slices airway lumen is separated from surrounding lung parenchyma by a wall composed from high-density vascular tissue. Due to the difference in tissue densities, an airway wall appears significantly brighter than the airway lumen and lung tissues. However airway lumen and lung parenchyma are both filled with air therefore on CT scan they appear as areas of very similar grey levels. In consequence only one pixel of airway wall discontinuousness causes the 3D region growing algorithm to leak into the lung area (see Fig. 3b). In practice broken airway walls appear frequently in case of lower order bronchi. It is caused by imperfections of imaging devices which results in loss of spatial resolution and increasing noxious influence of the noise content. Therefore it is almost impossible to segment whole airway tree with simple region growing and avoid the leakage.

5.2 Proposed Idea

The proposed airway tree segmentation algorithm, as most approaches to the considered problem, is based on seeded 3D region growing. However, during airway lumen segmentation region growing is applied twice. Moreover, leak prevention mechanism is applied in order to avoid the algorithm to consider voxels that are part of the lung parenchyma. Successive steps of the algorithm are as follows:

1. Image preprocessing.
2. Detection of main bronchi up to 5-6th divisions of airway tree using 3D seeded region growing.
3. Construction of a map defining possible locations of distal airways.
4. Detection of lower order bronchi using 3D region growing starting from previously detected airway tree and guided by the map of possible airway locations.

Step 1: *Image Preprocessing*

Before the main processing is applied, input CT data is smoothed in order to close broken airway walls and avoid leakages into the lungs. For data smoothing 3D median filter is applied. Traditional cubic (3×3×3) mask is used in this stage.

It should be underlined, that smoothing helps to avoid leakages but also compromises details by removing important airway data – especially airways of dimensions comparable with the kernel of the smoothing filter are cleared out.

Step 2: *First Pass of 3D Region Growing*

3D seeded region growing is applied for the first time on median-smoothed CT data. The seed is automatically defined on the first slice of the data set as a voxel located in the centre of oval area of the trachea. Applied in this stage 3D region growing algorithm classifies the current voxel as voxel belonging to airway lumen if both of the following constrains are true:

- intensity of the current voxel differs from the average intensity of voxels classified to airway lumen not more than *T%*;
- intensities of all 6 closest (connected) neighbours of the current voxel differ from the average intensity of voxels classified to airway lumen not more than *T%*.

Value of *T* is determined automatically on the first run. It equals to the lowest value for which region growing starts up. Then *T* is decreased by half after trachea is segmented in order to avoid leakages into the lungs. At this stage of airway tree segmentation up to 5-6th orders of bronchi can be detected.

Results of the first pass of region growing applied to two exemplary CT data sets are presented in Figure 4.

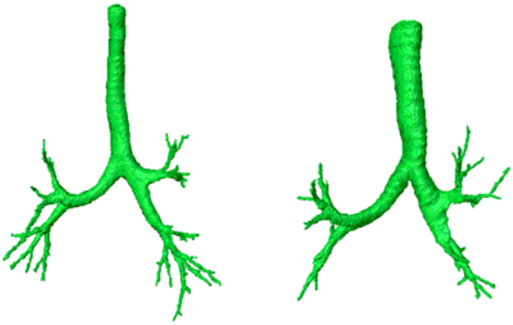


Fig. 4. Results of the first pass of region growing applied to two exemplary CT data sets

Step 3: *Construction of a map defining possible airway locations*
In the following step of the airway tree segmentation 3D morphological gradient is calculated based on non-smoothed CT data. This transformation (which is a difference between results of 3D greyscale dilatation and 3D greyscale erosion) highlights sharp grey levels transitions connected also (but not only) with airway walls. Areas of the highest gradient which are connected with airway tree built in the previous step of the algorithm define possible locations of the distal bronchi.

In order to define map of the possible candidate airways, gradient image is firstly thresholded and then eroded with small structural element. Airways are supposed to be located in those binary areas which are connected with airway tree built in the previous step. For segmentation (which defines the areas of the highest gradient) the authors applied thresholding with local iterative threshold selection [14].

It should be underlined that morphological gradient after thresholding is to define possible location of the distal bronchi. Therefore in order to avoid loss of important bronchi information the transformation should be performed on non-smoothed image.

Step 4: *Second Pass of 3D Region Growing*
The last step of the airway tree segmentation is a second application of the 3D region growing algorithm. This time however it is performed on the original (non-smoothed) CT data. The algorithm starts from previously segmented airway tree and is guided by the map constructed on the basis of morphological gradient. Successive voxels are joined to airway tree if both of the following constrains are fulfilled:

- current voxel is situated in the area defined by the map constructed on the basis of morphological gradient;
- intensity of the current voxel differs from the average intensity of voxels classified to airway lumen not more than $2T\%$.

Value of T is remembered from the second step of the algorithm and changes during algorithm performance in the way as during the first pass of region growing.

6 Results and Discussion

Results of complete airway tree segmentation procedure applied to previously used CT data sets are presented in Figure 5. Airways detected during the second application of region growing are marked in red color. The green color represents fragments of bronchial tree extracted using the first region growing step.

The assessment of the proposed algorithm was made by comparison of obtained airway trees with the corresponding templates. The templates were obtained by skilled radiologist who marked and filled airway lumen areas manually in successive slices of analyzed CT scans. Figure 6 presents the comparison between exemplary airway trees from Figure 5, which are representative of all tested data sets and the corresponding templates. Missing airways are marked in red color. Green color corresponds to the tree obtained using proposed algorithm.

One can see from Figure 6 that the proposed algorithm of airway tree segmentation enables to extract up to 8-9 generations of bronchi. Falsely detected airways characteristic for knowledge-based methods [7][8] do not appear at all. Moreover guided and constrained growth of an airway tree excludes possibility of leakages into the lungs.

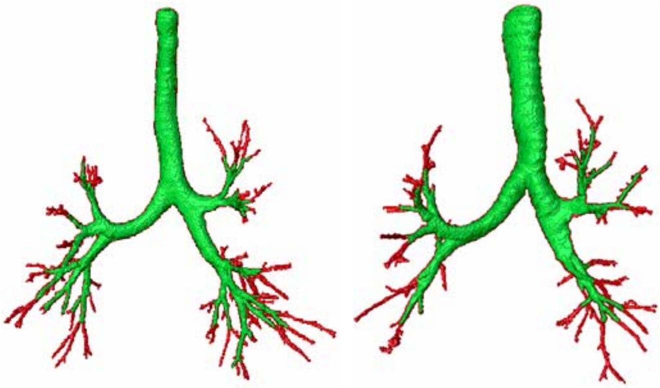


Fig. 5. Results of complete airway tree segmentation. Airways detected during second application of region growing are marked with red color.

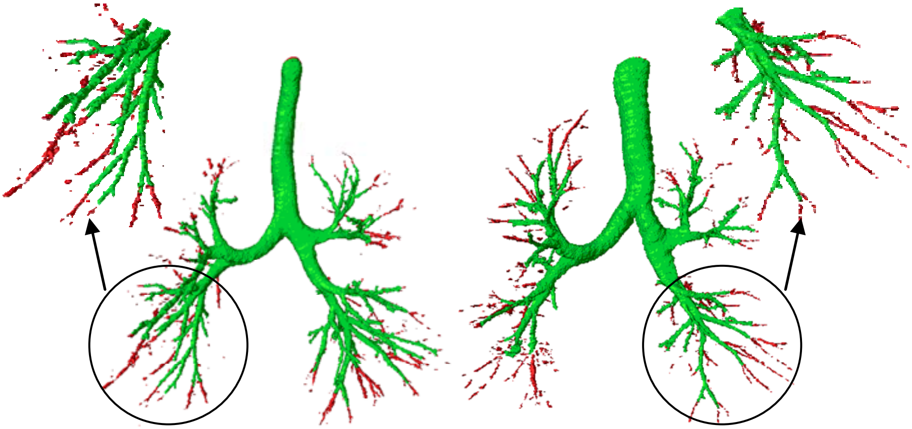


Fig. 6. Comparison of airway trees from Figure 5 with the corresponding templates. Missing airways are marked with red color.

The comparison between obtained airway trees and corresponding templates reveals that not more than one division of the bronchial tree was missed by the proposed algorithm. This means that information about 10 bronchi generations was present in considered CT data. This result is consistent with analysis presented in [9]. Having this in mind, the results presented in this paper can be considered interesting and accurate enough for further quantitative analysis of airway pathologies.

7 Conclusions

In this paper the problem of airway tree segmentation from CT chest scans was investigated. Especially fully automated algorithm for airway tree segmentation was introduced. The algorithm is based on a 3D approach which extracts airway trees from volumetric CT chest scans using region growing method guided and constrained by a morphological gradient. Such method allows to prevent leakages into the lungs and avoid falsely detected branches. Consequently, the proposed algorithm detects up to 9 generations of bronchi.

Nowadays, advanced CT scanners are able to resolve up to 10 orders of bronchial tree divisions in chest scans. Having this in mind the results obtained using the introduced algorithm can be considered interesting and accurate enough for further quantitative analysis of airway properties and its pathological changes.

References

1. American Thoracic Society: Standards for the diagnosis and care of patients with chronic obstructive pulmonary disease. Am. J. Resp. Crit. Care Med. 152, S77–S121 (1995)
2. Berger, P., Perot, V., Desbarats, P., Tunon-de-Lara, J.M., Marthan, R., Laurent, F.: Airway wall thickness in cigarette smokers: quantitative thin-section CT assessment. Radiology 235(3), 1055–1064 (2005)
3. Reilly, J.: Using computed tomographic scanning to advance understanding of chronic obstructive pulmonary disease. Proc. Am. Thorac. Soc. 3(5), 450–455 (2006)
4. Felita, C., Prêteux, F., Beigelman-Aubry, C., Grenier, P.: Pulmonary airways: 3-D reconstruction from multislice CT and clinical investigation. IEEE Trans. Med. Imag. 23(11), 1353–1364 (2004)
5. Graham, M., Gibbs, J., Higgins, W.: Robust system for human airway-tree segmentation. In: Proc. SPIE, vol. 6914, pp. 69141J-1 – 69141J-18 (2008)
6. Busayarat, S., Zrimec, T.: Detection of bronchopulmonary segments on high-resolution CT-preliminary results. In: 20th IEEE Int. Symp. Computer-Based Medical Systems, pp. 199–204 (2007)
7. Sonka, M., Park, W., Hoffman, E.: Rule-based detection of intrathoracic airway trees. IEEE Trans. Med. Imag. 15(3), 314–326 (1996)
8. Brown, M., McNitt, M., Mankovich, N., Goldin, J., Aberle, D.: Knowledge-based automated technique for measuring total lung volume from CT. In: Proc. SPIE, vol. 2709, pp. 63–74 (1996)
9. Aykac, D., Hoffman, E., McLennan, G., Reinhardt, J.: Segmentation and analysis of the human airway tree from three-dimensional X-ray CT images. IEEE Trans. Med. Imag. 22(8), 940–950 (2003)
10. Pisupati, C., Wolf, L., Mitzner, W., Zerhouni, E.: Segmentation of 3D pulmonary trees using mathematical morphology. In: Mathematical morphology and its applications to image and signal processing, pp. 409–416. Kluwer Academic Publishers, Dordrecht (1996)
11. Wood, S., Zerhouni, A., Hoffman, E., Mitzner, W.: Measurement of three-dimensional lung tree structures using computed tomography. J. Appl. Physiol. 79(5), 1687–1697 (1995)
12. Mayer, D., Bartz, D., Ley, S., Thust, S., Heussel, C., Kauczor, H., Straßer, W.: Segmentation and virtual exploration of tracheobronchial trees. In: 17th Int. Cong. and Exhibition Computer Aided Radiology and Surgery, London, UK, pp. 35–40 (2003)
13. Chabat, F., Xiao-Peng, H., Hansell, D., Guang-Zhong, Y.: ERS transform for the automated detection of bronchial abnormalities on CT of the lungs. IEEE Trans. Med. Imag. 20(9), 942–952 (2001)
14. Strzecha, K., Fabijańska, A., Sankowski, D.: Segmentation algorithms for industrial image quantitative analysis system. In: 17th IMEKO World Congress Metrology for a Sustainable Development, p. 164. Rio de Janeiro, Brazil (2006)

Geometric Approach to Hole Segmentation and Hole Closing in 3D Volumetric Objects

Marcin Janaszewski[1], Michel Couprie[2], and Laurent Babout[1]

[1] Computer Engineering Department,Technical University of Łódź,
Stefanowskiego 18/22, 90-924 Łódź, Poland
{janasz,lbabout}@kis.p.lodz.pl
[2] Université Paris-Est, LIGM, Equipe A3SI, ESIEE,
Cité DESCARTES BP 99 93162 Noisy le Grand CEDEX, France
coupriem@esiee.fr

Abstract. Hole segmentation (or hole filling) and hole closing in 3D volumetric objects, visualised in tomographic images, has many potential applications in material science and medicine. On the other hand there is no algorithm for hole segmentation in 3D volumetric objects as from the topological point of view a hole is not a 3D set. Therefore in the paper the authors present a new, geometrical approach to hole closing and hole filling in volumetric objects. Moreover an original and efficient, flexible algorithm of hole filling for volumetric objects is presented. The algorithm has been extensively tested on various types of 3D images. Some results of the algorithm application in material science for crack propagation analysis are also presented. The paper also includes discussion of the obtained results and the algorithm properties.

1 Introduction

From the topological point of view the presence of a *hole* in an object is detected whenever there is a closed path which cannot be transformed into a single point by a sequence of elementary local deformations inside the object [7]. For example, a sphere has no hole, a solid torus has one hole and a hollow torus has two holes. Unfortunately, from a topological point of view a hole is not a subset of 3D space so it can not be segmented or filled. On the other hand, there is strong practical need to treat holes as 3D subsets. In material science hole segmentation can contribute to the quantification of damage phenomena that can help to understand and further optimise the resistance of the material to damage [2,6]. Other possible medical applications may consist in filling small noisy holes in 3D tomographs of human organs. Such a hole filling is especially desired in analysis of 3D computer tomography images of bronchial tubes where noisy holes in a bronchial tree significantly complicate automatic quantitative analysis [8]. Therefore, taking into account the topological point of view we propose a geometrical approach, which considers the notion of the thickness of an object and interpolates the thickness in the corresponding hole filling volume.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 255–262, 2009.

2 Basic Notions

In this section, we recall some basic topological notions for binary images. A more extensive review is provided in [7,3].

We denote by $\mathbb{Z}$ the set of integers, $\mathbb{N}_+$ set of positive integers. Let $E = \mathbb{Z}^3$. Informally, a *simple point* p of a discrete object $X \subset E$ is a point which is "inessential" to the topology of X. In other words, we can remove the point p from X without "changing the topology of X".

Skipping some technical details, let $A(x, X)$ be the set of points of $X \setminus \{x\}$ lying in a neighborhood of x, and let $Ab(x, X)$ be the set of points of the complementary of X (background) lying in a neighborhood of x. Then, $T(x, X)$ (resp. $Tb(x, X)$) is the number of connected components of $A(x, X)$ (resp. $Ab(x, X)$). A point $x \in X$ is simple for X if and only if $T(x, X) = Tb(x, X) = 1$. Also, if a point $x \in X$ is such that $Tb(x, X) = 1$, then removing x from X does not create a new hole.

Let X be any finite subset of E. The subset Y of E is a *homotopic thinning of* X if $Y = X$ or if Y may be obtained from X by iterative deletion of simple points. We say that Y is an *ultimate homotopic skeleton of* X if Y is a homotopic thinning of X and if there is no simple point for Y.

Let $x \in E, r \in \mathbb{N}_+$, we denote by $B_r(x)$ the *ball of (squared) radius* r *centred on* x, defined by $B_r(x) \ = \{y \in E,\ d^2(x, y) < r\}$, where $d^2(x, y)$ is a squared Euclidean distance for any $x, y \in E$.

A ball $B_r(x) \subseteq X \subseteq E$ is *maximal for* X if it is not strictly included in any other ball included in X.

The *medial axis of* X, denoted by $\mathrm{MA}(X)$, is the set of the centres of all the maximal balls for X.

The *thickness of an object* X *in point* x belonging to $\mathrm{MA}(X)$ or to a skeleton of X is defined as a radius of the maximal ball centred in x.

3 Hole Notion from Geometrical Point of View

In our approach we consider holes from a geometrical point of view but we base our consideration on the topological definition of a hole presented in Sect. 1.

Intuitively, if we fill a hole in an object, the filling volume may be treated as a representation of the hole. In other words, the thickness (see Sect. 2) of the hole filling volume at its skeletal voxels should be equal to the thickness of the object at the skeletal voxels which are near the hole. Moreover, the shape of the boundary surface of a filling volume should fit the shape of the corresponding object's hole in the same way as two pieces of a puzzle match each other. The example of a hole filling volume for a frame is presented in Fig. 1. Unfortunately, it is very difficult to precisely define, in mathematical manner a hole filling volume. Taking into account the above comments, we are only able to propose two conditions which should be fulfilled by any segmented hole filling volume:

1. A hole filling volume should close a corresponding hole.

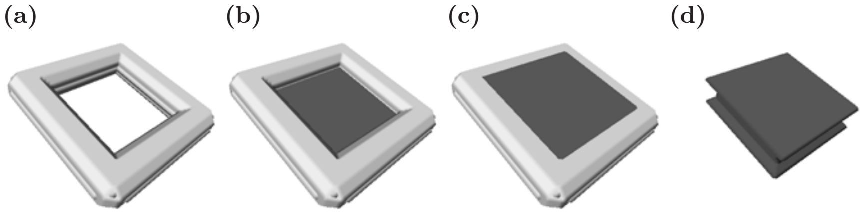


Fig. 1. An example of hole filling volume and hole closing patch for a frame: (a) isosurface of a frame; (b) frame with the hole closed; (c) frame with the hole filled. Hole filling volume is indicated with dark grey colour; (d) isosurface of the frame hole filling volume.

2. The thickness of a hole filling volume at its skeletal voxels should be equal to the thickness of the object on its medial axis (or skeletal) voxels which are close the hole.

4 Modified Hole Closing Algorithm

One of the main steps of hole filling algorithm (HFA) consists in application of a modified version of the hole closing algorithm (HCA) proposed by Aktouf et al. [1]. The original version of the algorithm is linear in time and space complexity and takes as an input volumetric objects and closes all holes in these objects with one voxel thick patches. The pseudocode of the algorithm can be presented as follows:

```
HCA ( Input $X$, Output $Y$)
  Generate a full cuboid $Y$ which contains $X$
  Repeat until no point to delete:
      Select a point $p$ of $Y \setminus X$ which is at the greatest distance from $X$ and
      such that $Tb(p, Y) = 1$
      $Y := Y \setminus p$
  Result: $Y$
```

An example of the algorithm result when applied to a 3D frame (Fig. 1(a))is presented in Fig. 1(b). It is worth mentioning the difference between hole filling and hole closing. The hole closing volume represented in dark grey colour in Fig. 1(b) is one voxel thick, independently on the thickness of the corresponding input object while the hole filling volume (see Fig. 1(c)) exactly matches the thickness of the corresponding object. More formal and extensive description of the algorithm can be found in [1].

The most important drawback of HCA, from the hole filling point of view, is that the shape of a hole closing patch may be significantly influenced by irrelevant branches which are close to the hole. Such a situation is presented in Fig. 2. In our approach HCA takes as an input a skeleton of a 3D object. The skeleton is one voxel thick so the object presented in Fig. 2(a) is a good example

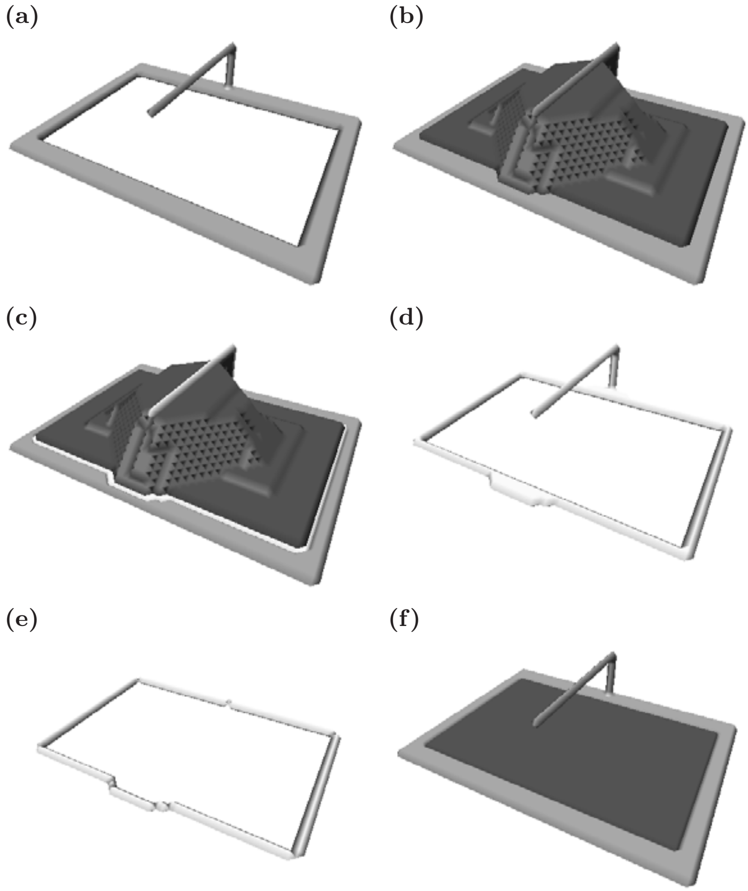


Fig. 2. Example result of each step of HCA+: (a) an isosurface of an input object; (b) the result of HCA. Notice that, the hole closing patch (dark grey colour) goes up to the branch situated over the hole; (c) the result of geodesic dilation of the patch over the input object. The intersection of dilated patch and input object, called hole contour (white colour); (d) visualisation of the hole contour, where one can see all its details (e) the result of ultimate homotopic skeletonisation algorithm applied to the hole contour. Note that, the branch has been deleted and topology of the hole contour has been preserved; (f) result of HCA, applied to the hole contour superimposed to the input object. Note that, the hole closing patch (dark grey colour) is not influenced by the branch.

.

of an input object for HCA. There is one big hole in the middle of the object and a thin branch above. Figure 2(b) presents the result of HCA. The hole closing patch, which is represented with dark grey colour, goes up to the branch, so it does not correspond to the "geometry of the hole", which leads to wrong hole filling. In this case we expect that the hole closing patch is flat as the object around the hole is flat. To overcome this problem we propose a 4-step method based on the HCA. The first step of the method consists in application of HCA.

As a result we obtain an object with the hole closed but the hole closing path is influenced by the branch. In the second step, the method realises only one iteration of geodesic dilation of the hole closing patch over the input object. An example showing the result of the dilation is presented in Fig. 2(c) where the intersection of a dilated hole closing patch and the input object is represented with white, shaded colour. In the following the intersection will be shortly called hole contour (see Fig. 2(d)). The third step consists in application of the ultimate homotopic skeleton algorithm [4] (UHSA) on the hole contour (see Fig. 2(e)). The last, fourth step consists in application of the HCA on the hole contour. As the hole contour does not contain any branch, the hole closing patch is not influenced by any branch and matches the geometry of the corresponding hole (see Fig. 2(f)).

This method, denoted by HCA+, can be computed in quasi-linear time as UHSA has quasi-linear time complexity and all other steps have linear time complexity [1,4].

5 Hole Filling Algorithm

In this section we propose the original hole filling algorithm (HFA), which consists of 4th main stages and can be presented in the following general pseudocode:

```
HFA ( Input $X$, $\theta$, Output $Z$)
    $S \leftarrow$ FES($X$, $\theta$)
    $P \leftarrow$ HCA+($S$)
    $P' \leftarrow$ MeanFilter($S$, $P$)
    $B \leftarrow$ DilationByBalls($P'$)
    $Z \leftarrow B - X$
```

The first step of HFA consists in application of **FES** procedure which generates the filtered Euclidean skeleton of an input object $X \subset E$ originally proposed by Couprie, Coeurjolly and Zrour [5]. This state-of-the-art algorithm for skeleton generation is based on well defined mathematical notions and allows to gradually prune a generated skeleton which makes the HFA resistant to small noisy branches and deformations of an input object. The second step: **HCA+** procedure has been described in details in Sect. 4. **Meanfilter** is a simple procedure which realises propagation of an object thickness represented by values of its filtered skeletal voxels, into hole closing patches. The algorithm, in each iteration, calculates a new value for each voxel, from a hole closing patch, as an average value of voxels from its neighbourhood which belong either to the hole closing patch or to the filtered skeleton. The algorithm stops when no significant changes occur during an iteration.

The last procedure: **DilationByBalls** for each voxel x of its input image generates a ball, centred in x, of radius equal to the value of voxel x.

Finally we obtain the hole filling algorithm which has the following properties:

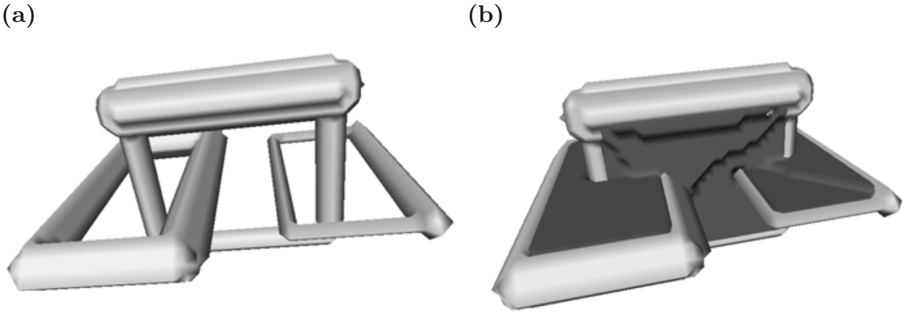


Fig. 3. Visualisation of a chain, whose links have different thicknesses. Hole filling volume is represented with dark grey colour: (a) an input object; (b) result of HFA applied to the chain.

- it is based on well defined mathematical notions and exact algorithms like: medial axis, bisector function, exact Euclidean distance, Euclidean skeleton,
- it generates volumes which fulfil both conditions 1 and 2 (see Sect. 3). The first one is guaranteed by **FES** and **HCA+** and the second is guaranteed thanks to **MeanFilter** and **DilationByBalls**,
- it is easy to use: only needs one parameter to be tuned (bisector threshold). Moreover it is easy to set the parameter as it belongs to the range $[0,\pi]$ and size of a hole closing patch changes monotonically with the bisector threshold. If the parameter is too small, then a skeleton of an input object is not enough pruned, hence noisy voxels from the surface of the object that could form cusp-like shapes may have influence on the hole filling volume which in that case is too thin. On the other hand, if the parameter is too large then an input object skeleton is over-pruned and the corresponding hole failing volume is too big and partly spread over the input object. Few tries are usually needed to set the proper bisector threshold for each input object.
- it is efficient: since most of its steps are optimal in time and space complexity.

Examples of the results of HCA are presented in Figs. 3, 4. In both examples, all holes are closed and the thickness of the corresponding hole filling volumes match the thickness of these objects, what is especially easy to observe in Fig. 3. So the two conditions about hole filling volume (see Sect. 3) are fulfilled. Figure 4(d) presents an example cross-section of the filled crack from Fig 4(c). It can be observed that the thickness of hole, its filling volume, corresponds to thickness of the crack. Moreover cross-section of hole closing is presented with light grey colour in the figure. The cross-section is one voxel thick and it is "centered" in the crack. For a material science point of view, the hole closing has a microstructural meaning since it represents so-called bridge ligaments, i.e. small area of special grain boundaries which present high resistance to cracking. This phenomenon is usually met in sensitised stainless steel that can undergo intergranular stress corrosion cracking when subjected to special corrosive media [2]. In that case, if

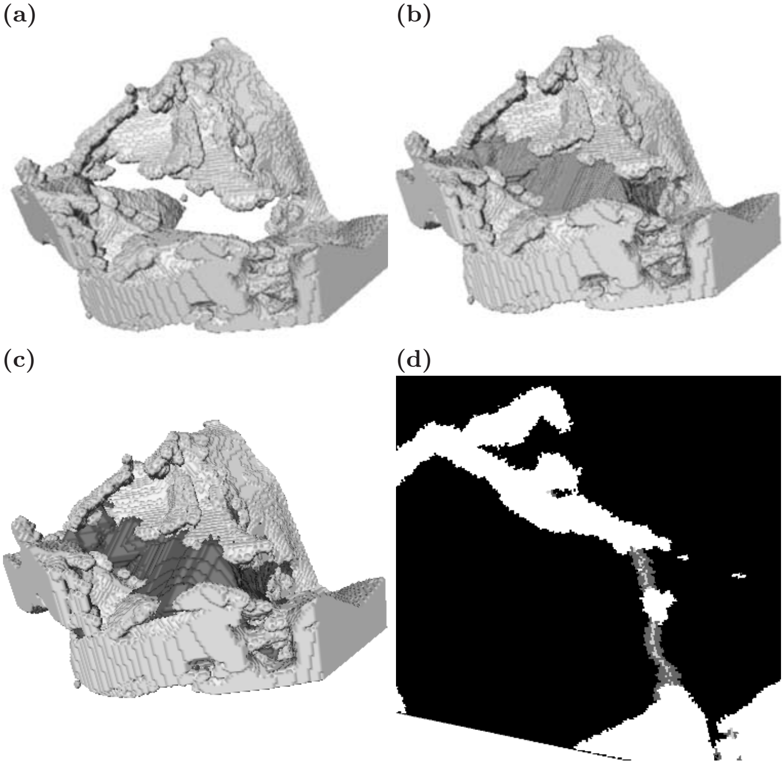


Fig. 4. A rendering of a crack path inside a material (material is represented by the background): (a) crack. Note that, there is a big hole inside the crack to be filled; (b) the crack with holes closed. Hole closing volume, one voxel thick, is represented with dark grey colour; (c) the crack with holes filled. Hole filling volume, which thickness corresponds to thickness of the crack, is represented with dark grey colour; (d) zoomed view of an oblique slice of the filled crack. Crack is represented with white colour, cross-section of hole filling volume is represented with dark grey colour and cross-section of hole closing volume is visualised with light grey colour.

a crack meets a bridge its branches go around the bridge and then merges. Hole filling has also a microstructural meaning since it is directly correlated to the local thickness of the crack portion that surrounds bridges. Work is currently in progress to correlate both morphological parameters of bridges, obtained after labelling of closed holes with local crack opening, retrieved from hole filling algorithm.

6 Conclusions

In the paper the authors have presented a flexible and efficient algorithm of hole filling for volumetric images. The algorithm has been tested on artificially constructed images and on images of a crack inside a material, for which it is an intended application. The visual analysis of results confirmed that the thickness of generated hole filling volumes correspond to the thickness of an input objects. According to our knowledge it is the first algorithm of hole filling for volumetric images.

Acknowledgements. The work is funded by the European Community's Sixth Framework Programme–Marie Curie Transfer of Knowledge Action (DENIDIA, contract No.: MTKD-CT-2006-039546). The work reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein.

The authors would like to thank Dr James Marrow (School of Materials, University of Manchester, UK) for permission to use volumetric images of crack.

References

1. Aktouf, Z., Bertrand, G., Perroton, L.: A three-dimensional holes closing algorithm. Pattern Recognit. Lett. 23, 523–531 (2002)
2. Babout, L., Marrow, T.J., Engelberg, D., Withers, P.J.: X-Ray microtomographic observation of intergranular stress corrosion cracking in sensitised austenitic stainless steel. Mater. Sci. Technol. 22, 1068–1075 (2006)
3. Bertrand, G.: Simple points, topological numbers and geodesic neighbourhoods in cubic grids. Pattern Recognit. Lett. 15, 1003–1011 (1994)
4. Bertrand, G., Couprie, M.: Transformations topologiques discretes. In: Coeurjolly, D., Montanvert, A., Chassery (eds.) Géométrie discrete et images numériques. J. M., Hermes, pp. 187–209 (2007)
5. Couprie, M., Coeurjolly, D., Zrour, R.: Discrete bisector function and Euclidean skeleton in 2D and 3D. Image and Vision Computing 25, 1543–1556 (2007)
6. King, A., Johnson, G., Engelberg, D., Ludwig, W., Marrow, J.: Observations of intergranular stress corrosion cracking in a grain-mapped polycrystal. Science 321(5887), 382–385 (2008)
7. Kong, T.Y., Rosenfeld, A.: Digital topology: introduction and survey. Comp. Vision, Graphics and Image Proc. 48, 357–393 (1989)
8. Park, W., Hoffman, E.A., Sonka, M.: Segmentation of intrathoracic airway trees: a fuzzy logic approach. IEEE Transactions on Medical Imaging 17, 489–497 (1998)

Optimizations and Performance of a Robotics Grasping Algorithm Described in Geometric Algebra

Florian Wörsdörfer[1], Florian Stock[2],
Eduardo Bayro-Corrochano[3], and Dietmar Hildenbrand[1]

[1] Technische Universität Darmstadt (Germany),
Graphical Interactive Systems Group
[2] Technische Universität Darmstadt (Germany),
Embedded Systems and Applications Group
[3] CINVESTAV Guadalajara (Mexico)

Abstract. The usage of Conformal Geometric Algebra leads to algorithms that can be formulated in a very clear and easy to grasp way. But it can also increase the performance of an implementation because of its capabilities to be computed in parallel. In this paper we show how a grasping algorithm for a robotic arm is accelerated using a Conformal Geometric Algebra formulation. The optimized C code is produced by the CGA framework Gaalop automatically. We compare this implementation with a CUDA implementation and an implementation that uses standard vector algebra.

Keywords: Conformal Geometric Algebra, Robot Grasping, CUDA, Runtime Performance.

1 Introduction

While points and vectors are normally used as basic geometric entities, in the 5D conformal geometric algebra we have a wider variety of basic objects. For example, spheres and circles are simply represented by algebraic objects. To represent a circle you only have to intersect two spheres, which can be done with a basic algebraic operation. Alternatively you can simply combine three points to obtain the circle through these three points. Similarly, transformations like rotations and translations can be expressed in an easy way. For more details please refer for instance to the book [4] as well as to the tutorials [7] and [5].

In a nutshell, geometric algebra offers a lot of expressive power to describe algorithms geometrically intuitive and compact. However, runtime performance of these algorithms was often a problem. In this paper, we investigate a geometric algebra algorithm of the grasping process of a robot [6] from the runtime performance point-of-view.

At first we present an alternative solution of the grasping algorithm using conventional mathematics and use its implementation as a reference for two

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 263–271, 2009.

optimization approaches. These approaches are based on Gaalop [8], a tool for the automatic optimization of geometric algebra algorithms. We use the optimized C code of Gaalop in order to compare it with our reference implementation. In the next step we implement this C code also on the new parallel CUDA platform [12] and compare the runtime performance with the other two implementations.

2 The Grasping Algorithm with Conventional Mathematics

In this chapter we give a description of the algorithm described below using standard vector algebra and matrix calculations. To keep this version comparable to the one using geometric algebra the same amount of work and time has been spend for both. We assume that all necessary data has been extracted from the stereo images as explained in Section 4.1.

The circle z_b is the circumscribed circle of the triangle Δ_b which is formed by the three base points. To compute its center two perpendicular bisectors have to be constructed. The intersection of them is the center point p_b of z_b. First the middle points $m_{12} = \frac{1}{2}(x_{b_1} + x_{b_2})$ and $m_{13} = \frac{1}{2}(x_{b_1} + x_{b_3})$ of two sides of Δ_b are computed.

Next the direction vectors $d_{12} = (x_{b_2} - x_{b_1}) \times n_b$ and $d_{13} = (x_{b_3} - x_{b_1}) \times n_b$ are needed to construct the perpendicular bisectors. For this the normal vector $n_b = (x_{b_2} - x_{b_1}) \times (x_{b_3} - x_{b_1})$ of the plane defined by the base points has to be constructed.

Now the perpendicular bisectors pb_{12} and pb_{13} and their intersection p_b can be computed:

$$p_b = m_{12} + \lambda_{12_S} \cdot d_{12} = m_{13} + \lambda_{13_S} \cdot d_{13} \tag{1}$$

Now we have everything we need to describe a circle in conventional mathematics except of the radius which is unnecessary for us in this case. To get an impression of one of the benefits of geometric algebra please compare all this steps to the first line of the listing in Figure 3 where a complete circle is constructed from three points using only one formula.

The circle z_b has to be translated in the direction of the normal vector n_b of the plane π_b in which z_b lies in. The distance z_b has to be translated is half the distance $d = \frac{n_b}{|n_b|}(x_a - p_b)$ between the point x_a and the plane π_b. So the translation vector is $T_b = \frac{1}{2}d \cdot \frac{n_b}{|n_b|}$. The normal vector of the plane in which z_t lies in equals the one of z_b, so $n_t = n_b$.

To be able to compute the necessary rotation the normal vector of the plane in which the gripper lies in has to be constructed. The robot is able to extract the center p_h of the gripper circle and two additional points g_1 and g_2 on it from the stereo pictures. With that the normal vector $n_h = (g_1 - p_h) \times (g_2 - p_h)$ of the gripper plane can be computed.

Because the needed rotation axes is perpendicular to the plane that is spanned by n_h and n_t its normal vector $n_{th} = n_h \times n_t$ has to be computed. With the vector n_{th} the rotation axes $l_R = l_R = p_h + \lambda \cdot n_{th}$ can be described.

The translation vector $l_T = p_t - p_h$ is just the difference of the two circle centers.

The angle between the two planes in which the circles lie in is equal to the angle between their normal vectors, so $\phi = acos\left(\frac{n_h \cdot n_t}{|n_h||n_t|}\right)$.

To perform the final rotation the following steps are necessary: compute the normalized rotation vector $n = \frac{n_{th}}{|n_{th}|}$, translate the rotation axes into the origin using RT_{orig}, compute the rotation using R and finally translate the axes back with RT_{back}.

$$RT_{orig} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_{h_1} & -p_{h_2} & -p_{h_3} & 1 \end{bmatrix} RT_{back} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_{h_1} & p_{h_2} & p_{h_3} & 1 \end{bmatrix} \qquad (2)$$

$$c = cos(\phi),\ s = sin(\phi),\ m = 1 - cos(\phi) \qquad (3)$$

$$R = \begin{bmatrix} n_1^2 m + c & n_1 n_2 m + n_3 s & n_1 n_3 m - n_2 s & 0 \\ n_1 n_2 m - n_3 s & n_2^2 m + c & n_2 n_3 m + n_1 s & 0 \\ n_1 n_3 m + n_2 s & n_2 n_3 m - n_1 s & n_3^2 m + c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

Finally the transformation can be computed by translating and rotating the points g_1 and g_2 from which the new position of the gripper circle can be derived.

3 Gaalop

The main goal of Gaalop is the combination of the elegance of algorithms using geometric algebra with the generation of efficient implementations.

Gaalop uses the symbolic computation functionality of Maple (together with a library for geometric algebras [1]) in order to optimize the geometric algebra algorithm developed visually with CLUCalc [13]. Gaalop computes the coefficients of the desired variable symbolically, returning an efficient implementation.

3.1 The Main Data Structure of Gaalop

The main data structure of Gaalop is an array of all the basic algebraic entities. They are called **blades** and are the basic computational elements and the basic geometric entities of geometric algebras. The 5D conformal geometric algebra consists of blades with **grades** 0, 1, 2, 3, 4 and 5, whereby a scalar is a **0-blade** (blade of grade 0). The element of grade five is called the pseudoscalar. A linear combination of blades is called a **k-vector**. So a bivector is a linear combination of blades with grade 2. Other k-vectors are vectors (grade 1), trivectors (grade 3) and quadvectors (grade 4). Furthermore, a linear combination of blades of different grades is called a **multivector**. Multivectors are the general elements of a geometric algebra. Table 1 lists all the 32 blades of conformal geometric algebra with all the indices as used by Gaalop. The indices indicate 1: scalar, 2 . . . 6: vector, 7 . . . 16: bivector, 17 . . . 26: trivector, 27 . . . 31: quadvector, 32: pseudoscalar.

Table 1. The 32 blades of the 5D conformal geometric algebra with the corresponding indices used by Gaalop

index	blade	grade
1	1	0
2	e_1	1
3	e_2	1
4	e_3	1
5	e_∞	1
6	e_0	1
7	$e_1 \wedge e_2$	2
8	$e_1 \wedge e_3$	2
9	$e_1 \wedge e_\infty$	2
10	$e_1 \wedge e_0$	2
11	$e_2 \wedge e_3$	2
12	$e_2 \wedge e_\infty$	2
13	$e_2 \wedge e_0$	2
14	$e_3 \wedge e_\infty$	2
15	$e_3 \wedge e_0$	2
16	$e_\infty \wedge e_0$	2
17	$e_1 \wedge e_2 \wedge e_3$	3
18	$e_1 \wedge e_2 \wedge e_\infty$	3
19	$e_1 \wedge e_2 \wedge e_0$	3
20	$e_1 \wedge e_3 \wedge e_\infty$	3
21	$e_1 \wedge e_3 \wedge e_0$	3
22	$e_1 \wedge e_\infty \wedge e_0$	3
23	$e_2 \wedge e_3 \wedge e_\infty$	3
24	$e_2 \wedge e_3 \wedge e_0$	3
25	$e_2 \wedge e_\infty \wedge e_0$	3
26	$e_3 \wedge e_\infty \wedge e_0$	3
27	$e_1 \wedge e_2 \wedge e_3 \wedge e_\infty$	4
28	$e_1 \wedge e_2 \wedge e_3 \wedge e_0$	4
29	$e_1 \wedge e_2 \wedge e_\infty \wedge e_0$	4
30	$e_1 \wedge e_3 \wedge e_\infty \wedge e_0$	4
31	$e_2 \wedge e_3 \wedge e_\infty \wedge e_0$	4
32	$e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \wedge e_0$	5

A point $P = x_1e_1 + x_2e_2 + x_3e_3 + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0$ for instance can be written in terms of a multivector as the following linear combination of blades

$$P = x_1 * blade[2] + x_2 * blade[3] + x_3 * blade[4] + \frac{1}{2}\mathbf{x}^2 * blade[5] + blade[6] \quad (5)$$

For more details please refer for instance to the book [4] as well as to the tutorials [7] and [5].

3.2 Use of Gaalop

For our application Gaalop is used in the following way: At first our algorithm is described using CLUCalc. One big advantage of CLUCalc is that the algorithm can be developed in an interactive and visual way. Another advantage is that the algorithm can be already verified and tested within CLUCalc. Gaalop uses this CLUCalc code in the next step in order to compute optimized multivectors using its symbolic optimization functionality. Leading question marks in the CLUCalc code indicate the variables that are to be optimized. Gaalop automatically generates C code for the computation of all the coefficients of the resulting multivectors. This C-Code is used as a basis for our CPU as well as our CUDA implementation.

4 The Algorithm in Geometric Algebra and Its Optimization

The algorithm used here is based on the robotics algorithm described in the paper [6]. It is used by the robot "'Geometer"' and can be downloaded as a CLUScript from [9].

4.1 Interface and Input Data

The algorithm needs four points that identify the object to be grasped. The robot acquires these points by taking a calibrated stereo pair of images of the object and extracting four non-coplanar points from these images.

After the points are gathered the orientation of the object has to be determined. For this the distance from one point to the plane spanned by the other three points is calculated. The point with the greatest distance d_a is called apex point x_a while the others are called base points $x_{b_1}, x_{b_2}, x_{b_3}$.

We assume that all the steps described above are already performed. So the starting point for our algorithm is the situation shown in Figure 1. The aim of the algorithm is now to compute the necessary translation and rotation for the gripper of the robot arm so that it moves to the middle of the base points and the apex point. This is done by first computing the grasping circle z_t as a translation of the base circle z_b. Then the necessary translator and rotor are computed to move the gripper circle z_h. The position of the gripper is also extracted from the stereo pictures by tracking its center and two screws on its rim.

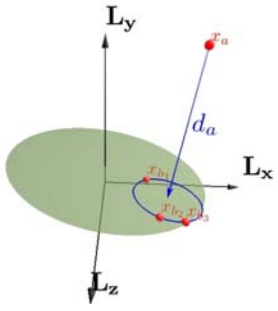


Fig. 1. The four points identifying the object to be grasped

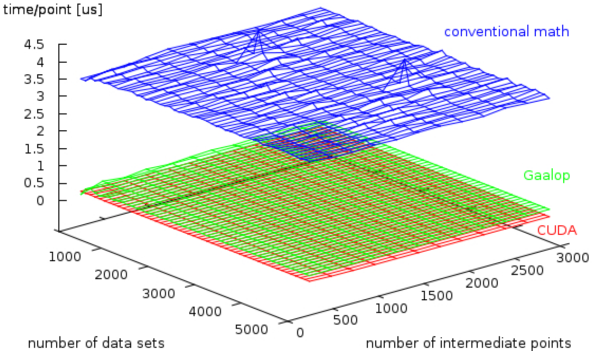


Fig. 2. Comparison of performance: The Gaalop code is up to fourteen times faster than the conventional math. The CUDA implementation gives a further performance improvement.

4.2 Short Description in Geometric Algebra

In this section we give a short description of the grasping algorithm using geometric algebra. The code of the listing in Figure 3 can be directly pasted into a CLUScript to visualize the results. The same code is used with Gaalop to produce the optimized C code. In our case we noticed that the `BasePlane` is needed as an intermediate expression to avoid the construction of extreme large expressions causing long computation times or sometimes even abnormal program termination. For the same reason the products `T*R` and `~R*~T` in the last line also were generated as intermediate results. Also Gaalop can only optimize the arguments of function calls like `abs` or `acos`.

```
zb_d = xb1 ^ xb2 ^ xb3;
?zb = *zb_d;
?BasePlane = *(zb_d ^ einf);
NVector = (BasePlane * einf).e0;
NLength = abs(NVector);
NVector = NVector/NLength;
Plane = BasePlane/NLength;
d_a=(xa.Plane)*NVector;
?T = 1 + 0.25 * d_a * einf;
?z_t = T * zb * ~T;
S_h = VecN3(g1,g2,g3)
      - 0.5*(g4*g4)*einf;
Pi_h = -e2;
?z_h = S_h ^ Pi_h;
s_h = -0.5*S_h*einf*S_h;
S_t = *z_t / ((*z_t) ^ einf);
s_t = -0.5*S_t*einf*S_t;
?l_T = s_h ^ s_t ^ einf;
?d = abs(l_T);
?l_T = l_T / d;
l_h = z_h ^ einf;
l_t = z_t ^ einf;
?Pi_th = l_t ^ (l_h*(einf^e0));
l_r_direct = s_h ^ (*Pi_th) ^ einf;
?l_r = *l_r_direct / abs(l_r_direct);
?phi = acos((l_t.l_h)
       / (abs(l_t)*abs(l_h)));
?R = cos(0.5*phi) - l_r*sin(0.5*phi);
?T = 1 + 0.5*d*l_T*einf;
?z_h_new = T*R*z_h*~R*~T;
```

Fig. 3. This is a CLUScript implementing the complete grasping algorithm. Only concrete values for the points of the object and the position of the gripper are missing.

First the base circle $z_b = x_{b_1} \wedge x_{b_2} \wedge x_{b_3}$ (the blue one depicted in Figure 1) has to be compute and translated in the direction and magnitude of $\frac{d_a}{2}$ with the translator $T_b = 1 + \frac{1}{4} d_a e_\infty$. This gives the target circle $z_t = T_b z_b \widetilde{T_b}$.

Now the necessary translation and rotation can be computed. To get the translator T the translation axes $l_T^* = p_h \wedge p_t \wedge e_\infty$ is needed. It is defined by the center points $p_t = z_t e_\infty z_t$ and $p_h = z_h e_\infty z_h$ of the two circles z_t and z_h. Also the distance between p_t and p_h has to be computed. It is given by $d = |l_T^*|$. Finally the translation is $T = 1 + \frac{1}{2} \Delta d \, l_T \, e_\infty$.

To compute the rotor R the axes of the circles z_t and z_h have to be used. They are $l_t^* = z_t \wedge e_\infty$ and $l_h^* = z_h \wedge e_\infty$ and are needed to calculate the plane $\pi_{th}^* = l_t^* \wedge (l_h^* e_0 \wedge e_\infty)$. This plane is used to get the rotation axes $l_r^* = p_h \wedge \pi_{th} \wedge e_\infty$. The angle between the two circles can be computed with the help of the inner product of their axes which gives $cos(\phi) = \frac{l_t^* \cdot l_h^*}{|l_t^*||l_h^*|}$. Finally the rotor is $R = e^{-\frac{1}{2} \Delta\, \phi\, l_r} = cos(\frac{1}{2} \Delta \phi) - l_r sin(\frac{1}{2} \Delta \phi)$.

4.3 C Code Generated by Gaalop

Because the C code Gaalop generates is quite large only a small portion of it will be given here. The following listing shows an excerpt of the C code for the first question mark from the listing above. For brevity `zb[8]` to `zb[16]` have been omitted.

```
float zb[32] = {0.0};
zb[7]=-xb3[4]*xb1[5]*xb2[6]+xb3[4]*xb1[6]*xb2[5]+xb2[4]*xb1[5]*xb3[6]
      -xb2[4]*xb1[6]*xb3[5]+xb1[4]*xb2[6]*xb3[5]-xb1[4]*xb2[5]*xb3[6];
...
```

5 The CUDA Implementation

General Purpose Graphics Processing Unit (GPGPU) are gaining much attention as cheap high performance computing platforms. They developed from specialized hardware, where general problems could only be computed by mapping the problem to the graphics domain, to versatile many-core processors. These offer, depending on the manufacturer, different programming models and interfaces. As our hardware is a NVIDIA based GTX 280 board, we use CUDA [12]. Other possibilities would be vendor independent RapidMind [11] or Brook++ [2], or the new OpenCL standard from the Khronos Group [10].

The GTX 280 consists of 30 multiprocessors, with each containing 8 parallel floating point data paths operating in a SIMD-like manner, i.e. they basically execute the same operation with different data. Are more threads scheduled the hardware automatically uses this to hide memory latencies.

As the computation for the grasping algorithm contains no control flow and access of the input pure sequential, further architectural particularities (i.e. memory organisation, control flow in parallel task) can be ignored. For a more detailed information about the architecture see the documentation [12].

The implementation of the grasping algorithm uses the same code for the computation as for the CPU, so the algorithm itself is not parallelized. The hundreds of parallel data paths are utilized by computing in each thread a single data set independent from all other threads.

In the following benchmarking, the host system was the benchmarked CPU system, and the measured times include the time to transfer the input onto the onboard memory and the time to transfer the result back to the host system.

6 Performance Measurements

Table 2 shows that the Gaalop code without any further manual improvements is up to fourteen times faster than the code of the conventional math algorithm although it seems to be sensitive to the quality of the input data. The reason for that is subject to further studies. Two third of the time needed by the implementation using conventional math is used for the calculation of the rotation matrices. All matrix calculations are done using the library `newmat`[3]. The CUDA implementation is more than three times faster than the Gaalop code.

Table 2. Time needed to compute a single intermediate point using a total of 1000 data sets with 240 intermediate points and the according speedup factor

Implementation	Time [μs]	Speedup
conventional math CPU	3.50	1
CGA CPU	0.25	14
CGA CUDA	0.08	44

In Figure 2 the time needed to compute one single intermediate step is plotted against the used number of data sets and the number of computed intermediate steps. It shows that the time needed to calculate one intermediate result is independent of the total number of intermediate steps and the number of used data sets for all implementations.

The performance measurements were conducted on a Pentium Core2Duo clocked at 3.06 GHz by taking the time to calculate a certain number of intermediate steps with a various number of random data sets. One data set consists of 3D-Cartesian coordinates for seven points. The algorithm can calculate an arbitrary number of intermediate results to represent the motion of the gripper.

7 Conclusion

In this paper, we compared three implementations of an algorithm for the grasping process of a robot from the performance point of view. The basic algorithm was developed using conventional vector mathematics. Then the geometrically very intuitive mathematical language of geometric algebra was used to develop a second version. This version was the basis for the parallelized CUDA implementation. It turned out that the geometric algebra algorithm when optimized with the Gaalop tool can be up to fourteen times faster than the conventional solution based on vector algebra. Another improvement can be achieved when implementing this optimized code on the parallel CUDA platform.

References

1. Ablamowicz, R., Fauser, B.: The homepage of the package Cliffordlib. HTML document (2005), http://math.tntech.edu/rafal/cliff9/ (last revised September 17, 2005)
2. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., Houston, M., Hanrahan, P.: Brook for gpus: Stream computing on graphics hardware. ACM Transactions on Graphics 23, 777–786 (2004)
3. Davies, R.B.: Newmat c++ matrix library. HTML document (2006), http://www.robertnz.net/nm_intro.htm
4. Dorst, L., Fontijne, D., Mann, S.: Geometric Algebra for Computer Science, An Object-Oriented Approach to Geometry. Morgan Kaufman, San Francisco (2007)
5. Hildenbrand, D.: Geometric computing in computer graphics using conformal geometric algebra. Computers & Graphics 29(5), 802–810 (2005)

6. Hildenbrand, D., Bayro-Corrochano, E., Zamora, J.: Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra. In: Advances in Applied Clifford Algebras. Birkhäuser, Basel (2008)
7. Hildenbrand, D., Fontijne, D., Perwass, C., Dorst, L.: Tutorial geometric algebra and its application to computer graphics. In: Eurographics conference Grenoble (2004)
8. Hildenbrand, D., Pitt, J.: The Gaalop home page. HTML document (2008), http://www.gaalop.de
9. Hildenbrand, D.: Home page. HTML document (2009), http://www.gris.informatik.tu-darmstadt.de/~dhilden/
10. Khronos Group. OpenCL Specification 1.0 (June 2008)
11. McCool, M.D.: Data-Parallel Programming on the Cell BE and the GPU using the RapidMind Development Platform, Rapidmind (2006)
12. NVIDIA Corp. NVIDIA CUDA Compute Unified Device Architecture – Programming Guide (June 2007)
13. Perwass, C.: The CLU home page. HTML document (2008), http://www.clucalc.info

Homological Computation Using Spanning Trees*

H. Molina-Abril[1,2] and P. Real[1]

[1] Departamento de Matematica Aplicada I, Universidad de Sevilla,
{real,habril}@us.es
http://ma1.eii.us.es/

[2] Vienna University of Technology, Faculty of Informatics, PRIP Group
habril@prip.tuwien.ac.at
http://www.prip.tuwien.ac.at/

Abstract. We introduce here a new $\mathbb{F}_2$ homology computation algorithm based on a generalization of the spanning tree technique on a finite 3-dimensional cell complex K embedded in $\mathbb{R}^3$. We demonstrate that the complexity of this algorithm is linear in the number of cells. In fact, this process computes an algebraic map ϕ over K, called homology gradient vector field (HGVF), from which it is possible to infer in a straightforward manner homological information like Euler characteristic, relative homology groups, representative cycles for homology generators, topological skeletons, Reeb graphs, cohomology algebra, higher (co)homology operations, etc. This process can be generalized to others coefficients, including the integers, and to higher dimension.

Keywords: Cell complex, chain homotopy, digital volume, homology, gradient vector field, tree, spanning tree.

1 Introduction

Homology (providing a segmentation of an object in terms of its n-dimensional holes) is one of the pillar of Topological Pattern Recognition. To compute homology for a nD digital object (with $n \geq 3$) is cubic in time with regards to the number n of cells [9,2,8]. Classical homology algorithms reduce the problem to Smith diagonalization, where the best available algorithms have supercubical complexity [12]. An alternative to these solutions are the reduction methods. They iteratively reduce the input data by a smaller one with the same homology, and compute the homolgy when no more reductions are possible [8,10].

To have at hand an algorithm computing homology in $O(n)$ is one of the main challenge in this area and has been recently conjectured in [8].

* This work has been partially supported by PAICYT research project FQM-296, "Andalusian research project" PO6-TIC-02268, Spanish MEC project MTM2006-03722 and the Austrian Science Fund under grant P20134-N13.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 272–278, 2009.

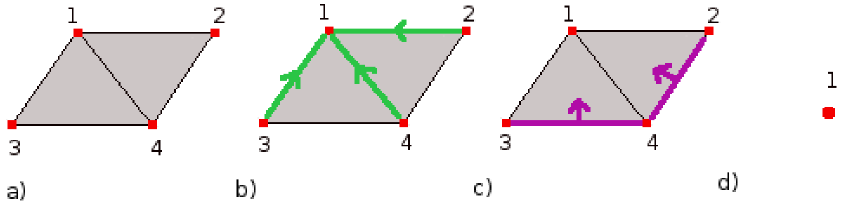


Fig. 1. (a) A cell complex K,(b) the first level of the forest determined by 0-cells and 1-cells, where $\phi(\langle 3\rangle) = \langle 1,3\rangle$, $\phi(\langle 4\rangle) = \langle 1,4\rangle$ and $\phi(\langle 2\rangle) = \langle 1,2\rangle$, (c) the second level of the forest determined by 1-cells and 2-cells, where $\phi(\langle 3,4\rangle) = \langle 1,3,4\rangle$ and $\phi(\langle 2,4\rangle) = \langle 1,2,4\rangle$, (d) $H_0(K) = \langle 1\rangle$

A finite cell complex K is a graded set formed of cells, with an operator ∂ describing the boundary of each cell in terms of linear combination of its faces. The finite linear combination (with coefficients in $\mathbb{F}_2 = \{0,1\}$) of cells form a graded vector space called chain complex associated to three dimensional cell complex K embedded in $\mathbb{R}^3$ and denoted by $C_*(K;\mathbb{F}_2)$. In [6] the solution to the homology computation problem (calculating n-dimensional holes) of K is described in the following terms: to find a concrete linear map $\phi : C_*(K;\mathbb{F}_2) \rightarrow C_{*+1}(K;\mathbb{F}_2)$, increasing the dimension by one and satisfying that $\phi\phi = 0$ (nilpotency condition), $\phi\partial\phi = \phi$ (chain contraction condition) and $\partial\phi\partial = \partial$ (cycle condition). In [5], a map ϕ of this kind is called homology gradient vector field (HGVF). This datum ϕ is, in fact, a chain homotopy operator on K (a purely homological algebra notion) and it is immediate to establish a strong algebraic link between the cell complex associate to K and its homology groups ($H_0(K)$, $H_1(K)$, $H_2(K)$).

In [7] the homological deformation process ϕ is codified to a minimal homological expression in terms of mixed trees. Different strategies for building these trees give rise to useful results in segmentation, analysis, topological skeleton, multiresolution analysis, etc. But the complexity of this solution for the homology computation problem is still cubic.

In this paper, we follow a different approach which allows to reduce the complexity of the problem. In the incidence graph of the cell complex K (in which the cells are represented by points and the (non-oriented) edges are determined by the relation "to be in the boundary of"), we perform a sort of spanning tree technique. This process gives as output a three-level forest (the first level determined by 0 and 1-cells, the second one by 1 and 2 cells, the third one by 2 and 3-cells).

A theoretical result will guarantee that considering some conditions during the generation of this forest, it can be seen as a HGVF. In this way the process for getting the homology generators of K starting from ϕ is $O(n)$ in time, where n is the number of cells of K.

In Section 2, we will show that a spanning forest for a 1-dimensional finite cell complex K gives raise to an HGVF $\phi : C_*(K) \rightarrow C_{*-1}(K)$. In Section 3 this result is extended to 3-dimensional finite cell complexes.

2 Spanning Trees as a Homology Gradient Vector Fields

Before presenting this new approach, some notions about algebraic topology must be introduced. A *q–chain* a of a three-dimensional cell complex K is a formal sum of cells of $K^{(q)}$ $(q = 0, 1, 2, 3)$. Let us consider the ground ring as the finite field $\mathbb{F}_2 = \{0, 1\}$. The q–chains form a group with respect to the component–wise addition; this group is the *qth chain complex* of K, denoted by $C_q(K)$. There is a chain group for every integer $q \geq 0$, but for a complex in $\mathbf{R}^3$, only the ones for $0 \leq q \leq 3$ may be non–trivial. The boundary map $\partial_q : C_q(K) \to C_{q-1}(K)$ applied to a q–cell σ gives us the collection of all its $(q-1)$–faces which is a $(q-1)$–chain. We say that $\sigma' \in \partial_q(\sigma)$ if σ' is a face of the q-cell σ. By linearity, the boundary operator ∂_q can be extended to q–chains, and satisfies $\partial_{q-1}\partial_q = 0$. From now on, a cell complex will be denoted by (K, ∂). A chain $a \in C_q(K)$ is called a *q–cycle* if $\partial_q(a) = 0$. If $a = \partial_{q+1}(a')$ for some $a' \in C_{q+1}(K)$ then a is called a *q–boundary*. Define the *qth homology group* to be the quotient group of q–cycles and q–boundaries, denoted by $H_q(K)$. For example in Figure 1, $\partial(\langle 2, 3, 4\rangle) = \langle 2, 3\rangle + \langle 2, 4\rangle + \langle 3, 4\rangle$, and the tree edges are faces of the 2-cell $\langle 2, 3, 4\rangle$. The 1-chain $\langle 2, 3\rangle + \langle 2, 4\rangle + \langle 3, 4\rangle$ is a 1–*cycle* and a 1–*boundary*.

Let (K, ∂) be a finite cell complex. A linear map of chains $\phi : C_*(K) \to C_{*+1}(K)$ is a *combinatorial gradient vector field* (or, shortly, combinatorial GVF) on K if the following conditions hold: (1) For any cell $a \in K_q$, $\phi(a)$ is a $q+1$-cell b; (2) $\phi^2 = 0$. Removing the first condition, ϕ will be called an *algebraic gradient vector field*. An algebraic GVF satisfying the conditions $\phi\partial\phi = \phi$ and $\partial\phi d = \partial$ will be called a *homology GVF* [6]. If ϕ is a combinatorial GVF which is only non-null for a unique cell $a \in K_q$ and satisfying the extra-condition $\phi\partial\phi = \phi$, then it is called a (combinatorial) *integral operator* [3]. An algebraic GVF ϕ is called *strongly nilpotent* if it satisfies the following property: Given any $u \in K^{(q)}$, if $\phi(u) = \sum_{i=1}^{r} v_i$ then $\phi(v_i) = 0$ for all $i = 1, \ldots, r$. We say that a linear map $f : C_*(K) \to C_*(K)$ is *strongly null over an algebraic gradient vector field* ϕ if given any $u \in K^{(q)}$, if $\phi(u) = \sum_{i=1}^{r} v_i$ then $f(v_i) = 0$ for all $i = 1, \ldots, r$.

Let (K, ∂) be a finite one dimensional cell complex (undirected graph) having only one connected component. The boundary operator $\partial : C_1(K) \to C_0(K)$ for a 1-cell e is given by $\partial(e) = v_2 + v_1$ (in $\mathbb{F}_2 = \{0, 1\}$), where v_1, v_2 are the endpoints of e. The boundary operator $\partial(w)$ for a 0-cell w is zero. Let $T = (V, E)$ a spanning tree (a tree composed of all the vertices) for K. Let us fix a root $v \in V$ for T and let us define the linear map $\phi : C_*(K) \to C_{*+1}(K)$ by

$\phi(w) = \{$the unique path from w to v through the tree $T\}$, $\forall w \in V$and
zero for every 1-cell of K.

In this definition, we understand by path a sum of edges in T connecting w with the root v. Then, the composition $\phi\phi$ is obviously zero and the conditions $\phi\partial\phi = \phi$ and $\partial\phi\partial = \partial$, where ∂ is the boundary operator for K, are also satisfied for every cell of K. In consequence:

Proposition 1. *The map ϕ described above determine a HGVF for the* 1-*dimensional cell complex K.*

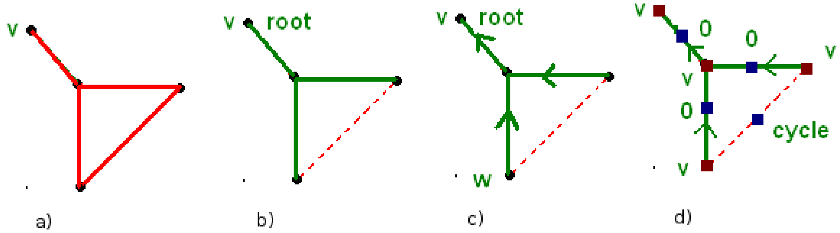


Fig. 2. (a) A graph K,(b) a spanning tree T, (c) description of ϕ, (d) description of π

Let $\pi : C_*(K) \to C_*(K)$ be a linear map defined by $\pi = id_{\mathcal{C}(\mathcal{K})} - \partial\phi - \phi\partial$. For each 0-cell w of K, $\pi(w) = v$. For each 1-cell e of K, $\pi(e) = 0$ if e belongs to T, and $\pi(e) = \{$ a representative cycle of a homology generator for $K\}$ if e does not belong to T. Let us consider now the incidence graph $IG(K)$ for K, that is, a graph with one vertex per point (red vertices forming the set V_r), one vertex per edge of K (blue vertices forming the set V_b) and an edge for every incidence between a point of V and a line belonging to E (see Figure 2). In other words, $IG(K)$ is the Hasse diagram [1] of the set of cells partially ordered by the relation of "to be a face of".The map π can be described as a function $\pi : V_r \bigcup V_b \to \operatorname{Ker}\partial$, which provides representative cycles of the different homology generators of K (evaluating π for those blue vertices not in T).

3 Homology Computation in Linear Time

Throughout this section, the extension of the previous spanning tree technique to higher dimensions is presented. A linear time algorithm for homology computation with coefficients in $\mathbb{F}_2$ is given.

Let (K, ∂) be a finite three-dimensional cell complex. Without loss of generality, suppose that K has only one connected component. Let consider the incidence graph $IG(K) = (V, E)$ for K, defined by the graph with one vertex per cell, and one edge for each incidence between an i-cell and an $i+1$-cell. The set of vertices and edges for $IG(V)$ can be decomposed in the following way:

$$V = \bigcup_{i=0}^{3} \{i\text{-cells for } K\}$$

$$E = \bigcup_{i=1}^{3} \{\text{unordered pairs } \{\sigma', \sigma\}, \text{ where } \sigma' \in \partial_i \sigma\}$$

Let $T^0 = (V^0, E^0)$ a tree spanning the vertices $V_0 = K_0$ in $IG(K)$. In T^0, $V^0 = V_0^0 \cup V_1^0$ (that we called, respectively, red and blue vertices of T^0), with $V_0^0 = V_0$, $V_1^0 \subset K_1$. Let us fix a red vertex ($v_0 \in K_0$) as the root of the tree T^0. Starting from the root v_0, let us obtain the maximum number of pairwise distints arcs whose tail is a red vertex and its head is a blue vertex. For doing this, we simply generate those arcs in T^0 from the edges composing the branches,

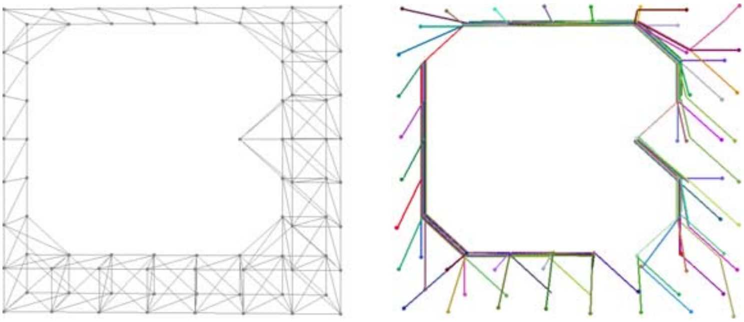


Fig. 3. (a)A 3-dimensional cell complex (b) T_0 tree

with tail being a red vertex and pointing them towards the root. Let us define $\phi_0(w)$ for a vertex w in K_0 by the sum of all the edges forming the unique path joining w with the root v_0 ($\phi_0(v_0)$ will be 0). It is straightforward to verify that $\phi_0 \partial_1 \phi_0 = \phi_0$ and $\partial_1 \phi_0 \partial_1 = \partial_1$. An example of the calculation of T^0 over a real $3D$-image is shown in Figure 3.

Now, we calculate a forest $\mathcal{F}^1 = (V^1, E^1)$ in $IG(K)$ spanning the vertices $V_0^1 = K_1 \setminus V_1^0$. In this graph, $V^1 = V_0^1 \cup V_1^1$ (red and blue vertices of $\mathcal{F}^1$), with $V_1^1 \subset K_2$ and it is constructed with the conditions: (a) given $e \in V_0^1$, all the 2-cell c, vertices of $IG(K)$ having as part of its boundary to e must be in V_1^1 as well the edge connecting c to e; (b) that if a 2-cell c is in V_1^1, then all the edges in $IG(K) \setminus E^0$ specifying those edges in K that are in the boundary of c, must be in $\mathcal{F}^1$ (see Figure 4).

Let us fix a red vertex $v_1^j \in V_0^1$ $(j = 1, \ldots, k)$ as a root for each of the trees $T_1^1, \ldots, T_m^1$ composing the forest $\mathcal{F}^1$. We only handle one of these tree T_1^1 and we do analogously for the others trees of $\mathcal{F}^1$. We first determine the red vertices in T_1^1 (1-cells in K) with degree greater or equal than three (they are called bifurcation red vertices). Among the 2-cells in V_0^1 having as a part of the boundary the red bifurcation vertex e, there will be a blue vertex $c_1(e)$ which is the parent of e and at least two more blue vertices $c_2(e), \ldots, c_r(e)$ which are the children of e in T_1^1. If v is a vertex of T_1^1, let us denote by $T_1^1(v)$ the subtree of T_1^1 generated by the descendants of v and their relationships in T_1^1 (including v as the root of $T_1^1(v)$). There is a semi-direct path from the red vertex w to the red vertex w' in T_1^1 if there is a sequence $w = w_1, z_1, \ldots, z_t, w' = w_{t+1}$ in which w_i are red vertices of T_1^1, z_i are blue vertices of T_1^1, (w_i, z_i) are arcs and $\{z_i, w_{i+1}\}$ are edges of T_1^1, for all $i = 1, \ldots, t$.

We now generate the maximum number of arcs (from the edges composing the branches) whose tail is a red vertex and pointing them away from the root. From this set, we eliminate those arcs (that is, we eliminate the arrow in the corresponding edge) that are associated to $n-1$ sons of a red bifurcation vertex of degree n. Let us define the map $\phi_1 : C_1(K) \to C_2(K)$. If $w \in V_1^0$, then $\phi_1(w) = 0$. If $w \in V_0^1$, then $\phi(w)$ is the sum of 2-cells (blue vertices in T_1^1) forming the different semi-directed pathes existing from w in $T_1^1(w)$.

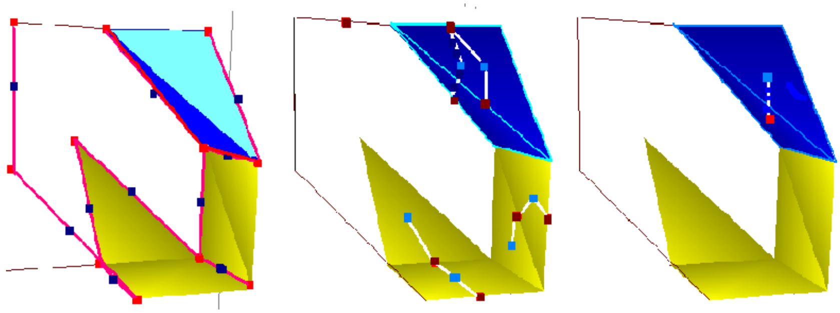


Fig. 4. (a) Spanning trees over the incidence graph of a 3-dimensional cell complex

This linear map verifies that $\phi_1\phi_0 = 0$, $\phi_1\partial_2\phi_1 = \phi_2$ and $\partial_2\phi_1\partial_2 = \partial_2$. The set of vertices $V_1^1 = \{e \in K_2 \ / \ e \in \phi_1(v)$, for some $v \in V_0^1\}$.

We now calculate a forest $\mathcal{F}^2 = (V^2, E^2)$ in $IG(K)$ spanning the vertices $V_0^2 = K_1 \setminus V_1^1$. In this graph, $V^2 = V_0^2 \cup V_1^2$ (red and blue vertices of $\mathcal{F}^2$), with $V_2^1 \subset K_3$ and it is constructed with the conditions: (a) given $e \in V_0^2$, all the 3-cell c, vertices of $IG(K)$, that have as part of its boundary to e must be in V_2^1 as well the edge connecting c to e; (b) that if a 3-cell c is in V_1^2, then all the edges in $IG(K) \setminus E^1$ specifying those 2-cells in K that are in the boundary of c, must be in $\mathcal{F}^2$ (see Figure 4). Using this forest, we define in an analogous way to $\mathcal{F}_1^1$, the map $\phi_2 : C_2(K) \to C_3(K)$. The set of vertices V_1^2 will agree with the set $\{e \in K_3 \ / \ e \in \phi_2(v)$, for some $v \in V_0^2\} \subset K_3$. Finally, ϕ_2 applied over an element of $K_3 \setminus V_1^2$ is zero.

The final map $\phi : C_*(K) \to C_{*+1}(K)$ satisfies the nilpotency, chain contraction and cycle conditions. The map $id_{\mathcal{C}(\mathcal{K})} + \partial\phi + \phi\partial$ applied to every leave of the corresponding tree provides us the different representative cycles for all the homology generators of K.

This process of $\mathbb{F}_2$-homology computation over a 3-dimensional cell complex, can be seen as the simple construction of three spanning trees but taking into account some special conditions. Considering a classical spanning tree technique as for example Depth-first search [13], which time complexity is $O(V + E)$ (V is the number of vertices of the graph and E the number of edges), the linearity of our method can be directly deduced.

4 Conclusions and Future Work

Many issues in computer imagery are related to the computation of homological information, like classification ([4] [11]), shape and pattern recognition ([10] [14]), etc. Image data require a huge amount of computational resources, and to find efficient algorithms which analyze image data is an active field of research. When dealing with 3–dimensional images, a fast computation is crucial, and it is even more with higher dimensionsal data.

A linear in time algorithm for computing homological information over a 3–dimensional cell complex is presented here. This method is based in spanning tree strategies. The main advantage of this result is its low computational time cost, in comparison with the complexity of the existing cubic in time methods.

There exist several spanning tree strategies. Some of them run in logarithmic time by using parallelization. Due to this fact, as future work, we plan to apply this parallelized methods to the construction of the homological forest in order to increase efficiency.

Another future aims is to deal with integer homology, instead of restricting the coefficients to $\mathbb{F}_2$, and to apply this method to different structures.

References

1. Birkhoff, G.: Lattice Theory. American Mathematical Society, Providence (1948)
2. Delfinado, C.J.A., Edelsbrunner, H.: An incremental algorithm for betti numbers of simplicial complexes on the 3–sphere. Comput. Aided Geom. Design 12, 771–784 (1995)
3. González-Diaz, R., Jiménez, M.J., Medrano, B., Molina-Abril, H., Real, P.: Integral operators for computing homology generators at any dimension. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 356–363. Springer, Heidelberg (2008)
4. Madjid, A., Corriveau, D., Ziou, D.: Morse homology descriptor for shape characterization. In: ICPR 2004: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR 2004), Washington, DC, USA, vol. 4, pp. 27–30. IEEE Computer Society, Los Alamitos (2004)
5. Molina-Abril, H., Real, P.: Advanced homological information on 3d digital volumes. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSSPR 2008. LNCS, vol. 5342, pp. 361–371. Springer, Heidelberg (2008)
6. Molina-Abril, H., Real, P.: Cell at-models for digital volumes. In: 7th IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition, Venice (Italy). LNCS, Springer, Heidelberg (2009)
7. Molina-Abril, H., Real, P.: Homological tree-based strategies for image analysis. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 326–333. Springer, Heidelberg (2009)
8. Mrozek, M., Pilarczykand, P., Zelazna, N.: Homology algorithm based on acyclic subspace. Computers and Mathematics with Applications 55, 2395–2412 (2008)
9. Munkres, J.: Elements of Algebraic Topology. Addison Wesley Co., Reading (1984)
10. Peltier, S., Ion, A., Kropatsch, W.G., Damiand, G., Haxhimusa, Y.: Directly computing the generators of image homology using graph pyramids. Image Vision Comput. 27(7), 846–853 (2009)
11. Scopigno, R., Zorin, D., Carlsson, G., Zomorodian, A., Collins, A., Guibas, L.: Persistence barcodes for shapes (2004)
12. Storjohann, A.: Near optimal algorithms for computing smith normal forms of integer matrices, pp. 267–274 (1996)
13. Tarjan, R.: Finding dominators in directed graphs. SIAM J. on Comput. 3, 62–89 (1974)
14. Zelawski, M.: Pattern recognition based on homology theory. MG&V 14(3), 309–324 (2005)

Getting Topological Information for a 80-Adjacency Doxel-Based 4D Volume through a Polytopal Cell Complex*

Ana Pacheco and Pedro Real

Dpto. Matematica Aplicada I, E.T.S.I. Informatica,
Universidad de Sevilla,
Avda. Reina Mercedes, s/n 41012 Sevilla (Spain)
{ampm,real}@us.es
http://ma1.eii.us.es/

Abstract. Given an 80-adjacency doxel-based digital four-dimensional hypervolume V, we construct here an associated oriented 4–dimensional polytopal cell complex $K(V)$, having the same integer homological information (that related to n-dimensional holes that object has) than V. This is the first step toward the construction of an algebraic-topological representation (AT-model) for V, which suitably codifies it mainly in terms of its homological information. This AT-model is especially suitable for global and local topological analysis of digital 4D images.

Keywords: 4–polytope, algebraic topological model, cartesian product, cell complex, integral operator, orientation.

1 Introduction

Homology (informing about 0, 1, 2 and 3–dimensional holes: connected components, "holes" or tunnels and cavities) of the 3D objects is an algebraic tool which allows to describe them in global structural terms [9]. This and others related topological invariants are suitable tools for some applications in which pattern recognition tasks based on topology are used. Roughly speaking, integer homology information for a subdivided 3D object (consisting in a collection of contractile "bricks" of different dimensionality which are glued in a "coherent" way) is described in this paper in terms of explicitly determining a boundary operator and a homology operator for any finite linear combination (with integer coefficients) of bricks such that, in particular, the boundary of the boundary (resp. the homology of the homology) is zero.

In [7], a method for computing homology aspects (with coefficients in the finite field $\mathbb{Z}/2\mathbb{Z} = \{0, 1\}$) of a three dimensional digital binary-valued volume V considered over a body-centered-cubic grid is described. The representation used

* This work has been partially supported by "Computational Topology and Applied Mathematics" PAICYT research project FQM-296, "Andalusian research project" PO6-TIC-02268, Spanish MEC project MTM2006-03722.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 279–286, 2009.

there for a digital image is an algebraic-topological model (AT-model) consisting in two parts: (a) (**geometric modeling level**) A cell complex $K(V)$ topologically equivalent to the original volume is constructed. A three dimensional cell complex consists of vertices (0–cells), edges (1–cells), faces (2–cells) and polyhedra (3–cells). In particular, each edge connects two vertices, each face is enclosed by a loop of edges, and each 3–cell is enclosed by an envelope of faces; (b) (**homology analysis level**) Homology information about $K(V)$ is codified in homological algebra terms [5,6]. This method has recently evolving to a technique which for generating a $\mathbb{Z}/2\mathbb{Z}$-coefficient AT-model for a 26–adjacency voxel-based digital binary volume V uses a polyhedral cell complex at geometric modeling level [11,12,17,19] and a chain homotopy map (described by a vector fields or by a discrete differential form) at homology analysis level [20,24]. Formally, an *AT-model* $((K(V), \partial), \phi)$ for the volume V can be geometrically specified by a cell (polyhedral) complex $K(V)$ and algebraically specified by a boundary $\partial : C(K(V))_* \rightarrow C(K(V))_{*-1}$ and a homology $\phi : C(K(V))_* \rightarrow C(K(V))_{*+1}$ operator, where $C(K(V))$ is the chain complex canonically associated to the polyhedral cell complex $K(V)$ (i.e., all the finite linear combinations of the elements of $K(V)$ are the elements of $C(K(V))$). These maps satisfy the following relations: (a) $\partial\partial = 0 = \phi\phi$; (b) $\phi\partial\phi = \phi$; (c) $\partial\phi\partial = \partial$. $K(V)$ is homologically equivalent (in fact, homeomorphically equivalent) to the voxel-based binary volume due to the fact that the process of construction of $K(V)$ is done in a local way by continuously deforming the geometric object formed by any configuration of "black" voxels (represented by unit cubes) in a $2 \times 2 \times 2$ neighborhood to the corresponding (polyhedral) convex hull of the barycenters of these voxels. In fact, the different cells of $K(V)$ are convex hulls of the configurations of m points placed in a $2 \times 2 \times 2$ elementary cube, with $m \leq 8$. The corresponding boundary and homology operators for each cell can be computed and saved in a look-up table for speeding up $\mathbb{Z}/2\mathbb{Z}$-homology runtime computation. This method is suitable for advanced topological analysis (computation of homology generators, Reeb graphs, cohomology rings . . .).

Homology with integer coefficients condenses the information provided by homology groups with coefficients in another commutative ring or field (like, the field of real numbers, the field of rational numbers, finite fields . . .). In [23], working with integer coefficients, a polyhedral $3D$ AT-model $(K(V), \partial), \phi)$ for a 26–adjacency voxel-based binary digital volume is constructed.

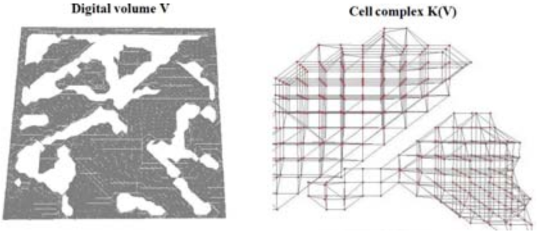


Fig. 1. Polyhedral cell complex $K(V)$ associated to a digital volume V

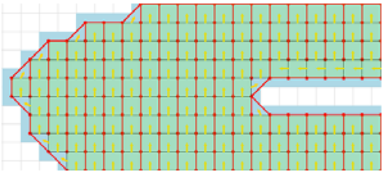


Fig. 2. Cell complex $K(V)$ associated to a 2D digital object and a visual description of a homology operator

In this paper, we work with a 4–dimensional ambiance (see [16]). More concretely, we work with a doxel-based digital binary 4D volume and using integer coefficients we determine a correct ("well oriented") global boundary operator $\partial_{K(V)}$ of the cell complex $K(V)$ as an alternating sum of the exterior faces of it. To do so, we construct $K(V)$ piece by piece specifying its corresponding local boundary operators knowing that they will be coherently glue one to each other to determine $\partial_{K(V)}$. A boundary isosurface extraction algorithm can be derived from this framework. Different homology computation techniques [2,3,6,7,15,22] can be applied to $K(V)$. Starting from $K(V)$ and using vector fields [20] or spanning-like trees [24,21], an algorithm (based on configuration look-up table) for constructing a global homology operator and, hence, a 4D AT-model of V appears as a feasible task and will be our objective in a near future.

2 4–Polytopal Continuous Analogous

We focus our interest in determining a orientation-correct 4–polytopal cell complex $K(V)$ topologically equivalent to V. The process to construct it is:

1. We divide the 4D–volume into overlapped (its intersection is a "3–cube" of eight mutually 80–adjacents doxels) unit hypercubes formed by sixteen mutually 80–adjacents doxels (see Figure 3). The different maximal cells of $K(V)$ will be suitable deformations of these unit hypercubes.

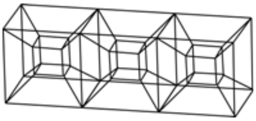


Fig. 3. Overlapped 2 × 2 × 2 × 2 hypercubes

2. We use cartesian product (CP) techniques to simplicially subdivide each unit 4D–cube as it is indicated in the Algorithm 1.

Algorithm 1. Obtaining a simplicialization using the CP

Let L_1, L_2, L_3, L_4 be 1-simplices and we consider the CP $L_1 \times L_2 \times L_3 \times L_4$. We can interpret the 0,1,2,3,4-simplices non-degenerate of the following way:

```
if $a$ is a non-degenerate 0-simplex then
  $a$ is vertex of the CP
else if $a$ is a non-degenerate 1-simplex then
  $a$ is an edge of the CP and it are obtained as follows:
  (a) The first element of the first 1-simplex ($a_{11}$), the first element of the second 1-simplex ($a_{12}$), the first element of the third 1-simplex ($a_{13}$) and the first element of the fourth 1-simplex ($a_{14}$) form in order the coordinates of the vertex ($a_{11}, a_{12}, a_{13}, a_{14}$) of the segment.
  (b) The second element of the first 1-simplex ($a_{21}$), the second element of the second 1-simplex ($a_{22}$), the second element of the third 1-simplex ($a_{23}$) and the second element of the fourth 1-simplex ($a_{24}$) form in order the coordinates of the vertex ($a_{21}, a_{22}, a_{23}, a_{24}$) of the segment.
else if $a$ is a non-degenerate 2-simplex then
  $a$ is a triangle whose vertices are $(a_{11}, a_{12}, a_{13}, a_{14}), (a_{21}, a_{22}, a_{23}, a_{24}), (a_{31}, a_{32}, a_{33}, a_{34})$
else if $a$ is a non-degenerate 3-simplex then
  $a$ is a tetrahedron and the coordinates of its vertices are obtained as above
else if $a$ is a non-degenerate 4-simplex then
  $a$ is a hypertetrahedron and the coordinates of its vertices are obtained as above
end if
```

3. (**cell deformation stage**) With each unit hypercube Q_4, we associate the corresponding 4–polytopal cell c and its border. The idea is to deform Q_4 using integral operators (elementary chain homotopy operators increasing the dimension by 1, which are non null only acting on one element [5]) to get the convex hull of this configuration. Now, we give an orientation to each cell c which preserves the global coherence on the cell complex (see Algorithm 2).

Algorithm 2. Obtaining the convex hull using integral operators

```
if $a$ is a white vertex then
  $\phi_{(a,b)}(a) = b$ where $b$ is an edge with $a$ as one of its vertices
else if $a$ is an edge with a white vertex then
  $\phi_{(a,b)}(a) = b$ where $b$ is an triangle with $a$ as one of its vertices
else if $a$ is a triangle with a white vertex then
  $\phi_{(a,b)}(a) = b$ where $b$ is an tetrahedron with $a$ as one of its vertices
else if $a$ is a tetrahedron with a white vertex then
  $\phi_{(a,b)}(a) = b$ where $b$ is an hypertetrahedron with $a$ as one of its vertices
end if
```

Remark 1. Let us note that a vertex is called black if it belongs to the initial object, otherwise the vertex is white.

To finish this section, we are going to highlight “good” properties of our four-dimensional model: (a) It can capture the homology at the same time that we construct it; (b) It allows us to render the boundary surface of the 4–polytopal continuous analogous $K(V)$; (c) It can be generalized to nD.

3 Local Convex Hulls for the 4–Polytopal Cell Complex

In this section we show a simplicial decomposition of the elementary hypercube Q_4. This decomposition will help us in determining the correct boundary operator for the deformed cells coming from the different configurations of black vertices in the standard unit $4D$–cube.

To represent a set of vertices of Q_4, we use two different visualizations:

1. **(by 3D slices)** We consider the $4D$ object divided into $3D$ slices, such an object may be thought of as a "time series of $3D$ objects" (see Figure 4).

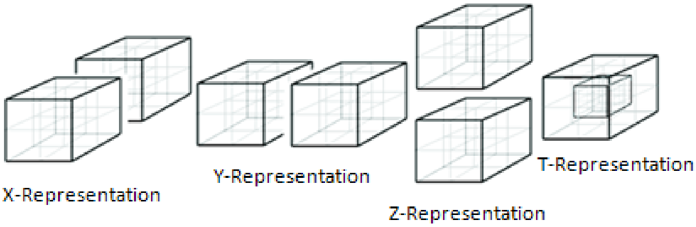


Fig. 4. Visualizing a $4D$ object in $3D$ slices

2. **(by Schlegel diagram)** It consists on a projection of a polytope, from a n–dimensional space into $(n-1)$–dimensional space, through a point beyond one of its facets. It is also called tesseract (see Figure 5).

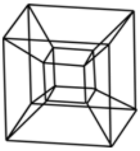


Fig. 5. Visualizing a $4D$ object using the Schlegel diagram

Using the first visualization (for example in the Y-Representation), in order to obtain a Q_4 simplicialization (see Figure 4), we must compute the barycenter of each one of the eight 3–cubes which form the boundary of the unit 4–cube and so we will get two new cubes (see Figure 6) which we must simplicialize using Algorithm 1; so we will have the Y-Representation of a Q_4 simplicialization.

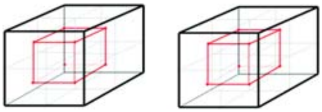


Fig. 6. Obtaining a simplicialization of Q_4 by $3D$ slices

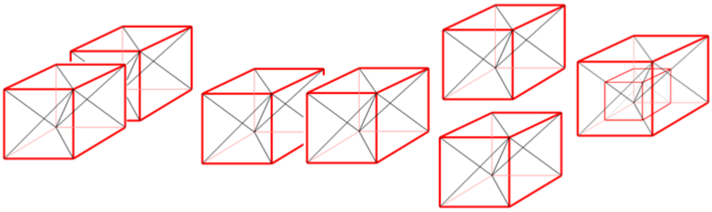


Fig. 7. X,Y,Z,T-Representation of a Q_4 simplicialization

The same way, we can obtain X,Z,T-Representation of a Q_4 simplicialization.

In order to visualize the simplicialization of the interior of Q_4 we use the tesseract visualization (see Figure 5).

First of all, we need to know pentatopes (hypertetrahedra) in which Q_4 is decomposed. To do this, we use the degeneracy operators of the CP. In this way, we obtain the 24 pentatopes of the hypercube Q_4.

Now, we have to order the vertices of the pentatopes in such a way that each one has inverse orientation to its neighbors. The 4–cube Q_4 is then defined as a cell complex, since the orientation of its pentatopes allows us to determine a correct boundary operator.

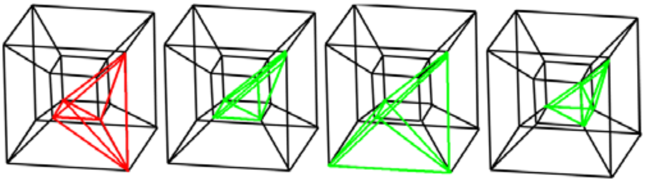


Fig. 8. HT_3 (in red) with its neighbors

Finally, we show here an example for getting the final boundary operator for a 4–polytopal cell, applying integral operators to the unit 4–cube Q_4.

We suppose that we have a unit 4–cube configuration of 15 vertices, without loss of generality, we can suppose that the vertex $(1, 0, 0, 1)$ is removed. Indeed, it is equivalent to say that we have a unit 4–cube configuration where 15 of them are black and one of them is white.

Using the Algorithm 2 we must define the following integral operators for obtaining the convex hull of the configuration (affected simplices in Figure 9):

$\phi_{((1,0,0,1),<(1,0,0,1),(1,0,1,1)>)}$

$\phi_{((1,0,0,1),<(1,0,0,1),(1,1,0,1)>)}$

$\phi_{((1,0,0,1),<(1,0,0,1),(0,0,0,1)>)}$

$\phi_{(<(1,0,0,1),(0,0,0,0)>,<(1,0,0,1),(0,0,0,0),(1,0,1,1)>)}$

$\phi_{(<(1,0,0,1),(1,1,1,1)>,<(1,0,0,1),(1,1,1,1),(0,0,0,0)>)}$

$\phi_{(<(1,0,0,1),(0,0,0,0),(1,0,1,1)>,<(1,0,0,1),(0,0,0,0),(1,0,1,1),(1,0,0,0)>)}$

$\phi_{(<(1,0,0,1),(0,0,0,0),(1,1,1,1)>,<(1,0,0,1),(0,0,0,0),(1,1,1,1),(1,0,1,1)>)}$

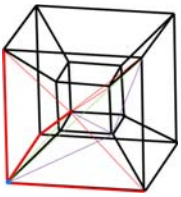


Fig. 9. Integral operators acting on Q_4: In blue the vertex affected, in green the edges affected, in red the triangles affected and in purple the tetrahedron affected

$$\phi(<(1,0,0,1),(0,0,0,1),(1,1,1,1)>,<(1,0,0,1),(0,0,0,1),(1,1,1,1),(1,1,0,1)>)$$
$$\phi(<(1,0,0,1),(0,0,0,0),(1,1,0,1)>,<(1,0,0,1),(0,0,0,0),(1,1,0,1),(1,1,1,1)>)$$
$$\phi(<(1,0,0,1),(0,0,0,0),(1,1,1,1),(1,1,0,0)>,<(1,0,0,1),(0,0,0,0),(1,1,1,1),(1,1,0,0),(0,0,1,0)>)$$

4 Conclusions and Applications

This paper is a step toward the extension to $4D$ and integer coefficients of the $3D$ AT-model proposed in [19,20]. Given a binary doxel-based $4D$ digital object V with 80–adjacency relation between doxels, it is possible to construct a homologically equivalent oriented hyperpolyhedral complex $K(V)$ in linear time. Starting from this result, it is possible to design an algorithm for computing homology information of the object. In this algorithm, a look-up table with all the possible configurations of black doxels in the unit 4–cube Q_4 (that is, all possible polytopal unit cells) saves their boundary operator, simplicialization and homology operator. In a near future, we will intend to develop a technique for homology computation of $4D$ digital objects based on this schema.

References

1. Couprie, M., Bertrand, G.: New Characterizations of Simple Points in 2D, 3D and 4D Discrete Spaces. IEEE Trans. Pattern Analysis and Machine Intelligence 31, 637–648 (2009)
2. Delfinado, C.J.A., Edelsbrunner, H.: An Incremental Algorithm for Betti Numbers of Simplicial Complexes on the 3–Sphere. Computer Aided Geometric Design 12, 771–784 (1995)
3. Dey, T.K., Guha, S.: Computing Homology Groups of Simplicial Complexes in $\mathbb{R}^3$. Journal of the ACM 45, 266–287 (1998)
4. Forman, R.: A Discrete Morse Theory for Cell Complexes. In: Yau, S.T. (ed.) Geometry, Topology & Physics for Raoul Bott, pp. 112–125. International Press (1995)
5. Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B., Molina-Abril, H., Real, P.: Integral Operators for Computing Homology Generators at Any Dimension. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 356–363. Springer, Heidelberg (2008)
6. Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B., Real, P.: Chain homotopies for object topological representations. Discrete Applied Mathematics 157, 490–499 (2009)

7. Gonzalez-Diaz, R., Real, P.: On the cohomology of 3D digital images. Discrete Applied Mathematics 147, 245–263 (2005)
8. Gonzalez-Diaz, R., Medrano, B., Real, P., Sanchez-Pelaez, J.: Algebraic Topological Analysis of Time-sequence of Digital Images. In: Ganzha, V.G., Mayr, E.W., Vorozhtsov, E.V. (eds.) CASC 2005. LNCS, vol. 3718, pp. 208–219. Springer, Heidelberg (2005)
9. Hatcher, A.: Algebraic Topology. Cambridge University Press, Cambridge (2001)
10. Kenmochi, Y., Imiya, A.: Discrete Polyhedrization of a Lattice Point Set. In: Bertrand, G., Imiya, A., Klette, R. (eds.) Digital and Image Geometry. LNCS, vol. 2243, pp. 150–162. Springer, Heidelberg (2002)
11. Kenmochi, Y., Imiya, A., Ichikawa, A.: Discrete Combinatorial Geometry. LNCS, vol. 30, pp. 1719–1728. Springer, Heidelberg (1997)
12. Kenmochi, Y., Imiya, A., Ichikawa, A.: Boundary Extraction of Discrete Objects. Computer Vision and Image Understanding 71, 281–293 (1998)
13. Kong, T.Y., Roscoe, A.W., Rosenfeld, A.: Concepts of Digital Topology. Topology and its Applications 46, 219–262 (1992)
14. Kovalevsky, V.A.: Finite Topology as Applied to Image Analysis. Computer Vision, Graphics, and Image Processing 46, 141–161 (1989)
15. Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational homology. Series Applied Mathematical Sciences, vol. 157. Springer, Heidelberg (2004)
16. Lee, M., De Floriani, L., Samet, H.: Constant-Time Navigation in Four-Dimensional Nested Simplicial Meshes. In: Proc. International Conference on Shape Modeling and Applications, pp. 221–230 (2004)
17. Mari, J.L., Real, P.: Simplicialization of Digital Volumes in 26-Adjacency: Application to Topological Analysis. Pattern Recognition and Image Analysis 19, 231–238 (2009)
18. May, J.P.: Simplicial Objects in Algebraic Topology. University of Chicago Press, Chicago (1967)
19. Molina-Abril, H., Real, P.: Advanced homological computation of digital volumes via cell complexes. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 361–371. Springer, Heidelberg (2008)
20. Molina-Abril, H., Real, P.: Cell AT-models for digital volumes. In: Torsello, A., Fscolano, F., Brun, L. (eds.) GBRPR 2009. LNCS, vol. 5534, pp. 314–323. Springer, Heidelberg (2009)
21. Molina-Abril, H., Real, P.: Homological Computation using Spanning Trees. In: CIARP 2009, Guadalajara, Mexico (2009)
22. Mrozek, M., Pilarczykand, P., Zelazna, N.: Homology algorithm based on acyclic subspace. Computers and Mathematics with Applications 55, 2395–2412 (2008)
23. Pacheco, A., Real, P.: Polyhedrization, homology and orientation. In: Wiederhold, P., Barneva, R.P. (eds.) Progress in Combinatorial Image Analysis, pp. 153–167. Research Publishing Services (2009)
24. Real, P., Molina-Abril, H., Kropatsch, W.: Homological tree-based strategies for image analysis. In: Computer Analysis and Image Patterns, CAIP (2009) (accepted)
25. Skapin, X., Lienhardt, P.: Using Cartesian Product for Animation. Journal of Visualization and Computer Animation 12, 131–144 (2001)

Circular Degree Hough Transform

Alejandro Flores-Mendez and Angeles Suarez-Cervantes

Universidad la Salle,
LIDETEA,
México D.F., México
aflores@ci.ulsa.mx, msuarez@lci.ulsa.mx
http://www.ci.ulsa.mx/~aflores

Abstract. The Circular Hough Transform (CHT) is probably the most widely used technique for detecting circular shapes within an image. This paper presents a novel variation of CHT which we call the Circular Degree Hough Transform (CDHT). The CDHT showed better performance than CHT for a number of experiments (eye localization, crater detection, etc.) included in this document. The improvement is mainly achieved by considering the orientation of the edges detected.

Keywords: Hough Transform, Circle Detection.

1 Introduction

One of the main problems in the image processing area refers to the detection of common shapes such as lines and circles. For this purpose, a number of techniques have been developed raging from very general ones to heuristical *ad hoc* algorithms [1]. Among the former, we can find the Hough Transform [2] which can detect analytically defined shapes. A decade after the appearance of the Hough Transform (HT), Duda and Hart [3] developed what nowadays is called the Generalized HT (GHT) which can be used to detect arbitray shapes not necessarily having a simple analytical form. From the GHT, a number of variations have been developed, specially for the case of circle detection [4], [5], [6], [7], [8] which is the case we are interested in this paper. Besides these variations, *Thresholding*, *Sliding Window*, etc. have been used for proper detection. These variations are necessary since only in scarce occasions the circular shape we want to detect in an image truly corresponds to a circle, and even if the image contains a circle it might not be complete. On the other hand, the number of applications that could benefit from the adequate detection of circular shapes count in great numbers. Some examples are crater detection[9], eye localization[10], RoboCup [11], just to name a few.

The current paper shows a novel method, which uses the information related to the edges of an image. In fact, this proposal uses the degree or orientation of the edges detected. However, instead of "tracing a line" into the accumulator in the direction related to the edges [7][1], we use these as a parameter to maximize the elements of the accumulator for which the likelihood between the ideal degree and the edges degree is attained.

[1] Which often causes problems, even for small degree errors.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 287–294, 2009.

The paper is organized as follows. Section 2 describes the proposed algorithm in detail. Section 3 presents some experimental results. In Section 4 the conclusions are included.

2 The Circular Degree Hough Transform

Before we detail how is it that the Circular Degree Hough Transform (CDHT) works, we will briefly summarize how is that the CHT does. Basically, the CHT algorithm obtains the edges of an image. From them, it calculates the *accumulator*, which is a matrix equivalent to the result of the correlation between these edges and a mask containing a circle. Thus, the accumulator holds evidence of the center coordinates where a circle of a given radius r could be.

Canny is commonly the selected technique to detect the edges of an image, because it is relatively immune to noise as it includes a Gaussian filter. Moreover, Canny gives a single response for a single edge. This is important when trying to locate a circular shape since this reduces spurious responses that could affect what is stored in the accumulator. Now, let `edges` be a vector whose elements are the coordinates of the edges detected. Then, the corresponding pseudocode for CHT follows:

```
CHT (edges, r, ang_dif)
initialize (acum)
for i ← 1 to length [edges]
    (x, y) ← edges [i]
    $\alpha$ ← 0
    repeat
        $\alpha$ ← $\alpha$ + ang_dif
        a ← discretized (x - r cos ($\alpha$))
        b ← discretized (y - r sin ($\alpha$))
        if (range (a, b))
            acum [a, b] ++
    until $\alpha > 2\pi$
return acum
```

where `range(a,b)` determines if `a,b` are proper values to index the `acum` variable. Notice that the maximums of the accumulator are the best candidates to be the center of a circular shape. The process is depicted in Fig. 1.

A variation of this process compares the elements of `acum` against a threshold, θ to determine if a circle is really included in the image[2]. Yet another variation, increases the values of `acum` not only for `(a,b)`, but for a neighborhood of this coordinate. This variation is called *Sliding Window* and is commonly used when the circle in the image has suffer a transformation into a slightly eccentric ellipse.

A problem with these approaches is that they are highly sensitive to noise, since many spurious edges could be detected. Moreover, in some cases[9], the

[2] The circles detected will be those elements for which `acum` $> \theta$. For some cases, before the comparison, it is convenient to normalize `acum` to the range $[0, 1]$.

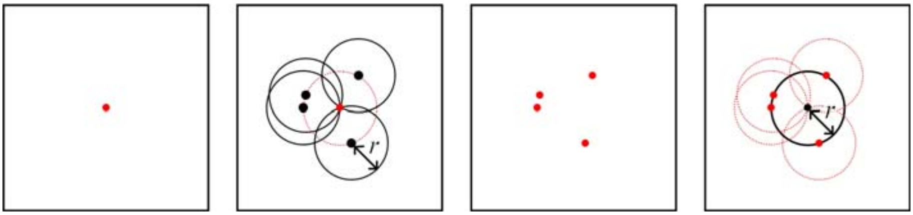


Fig. 1. (From left to right) 1) A point in an image 2) is used to determine if a circle of a given radius r contains it. This is stored in the accumulator (black points). 3) When the edges are calculated, the same process is done for every point on the image, 4) and the values of the accumulator (that hold the number of 'intersections') are used to determine the center of a circle with radius r within the image.

image itself contains features that are not noise, but will impact the edge detection process. In fact, if there are numerous edges not necessarily related to a circular shape, then the CHT could return a number of *false* detections. To avoid this, the image could be preprocessed by using some kind of filtering, but this is hardly useful if the edges are not noise related.

Pending on the application, a number of solutions could be helpful. For example, in iris detection, before the accumulator is increased, it could be checked that the edge separates a dark object of "adequate" dimensions surrounded by a bright object. Another modification could also consider the magnitude of the edges detected. The problem with these approaches is that they will be more noise resilient for some kinds of noise (even if no previous filtering is considered), but the numerous edges problem remains.

To solve this, it is possible to take into account the geometrical restrictions of the circular shapes. A simple manner to include this is to consider the angle of the edges. In fact, this was already explored [6], [7]. In these papers, the edge direction is used to "trace a line" along the direction of the edge detected, altering in this way the accumulator. In other words, these variations calculate the edges, obtain the directions associated to them, and increase the elements of the accumulator along the direction detected. The center of the circular shapes appearing on the image corresponds to those elements where the lines intersect (this process is depicted in Fig. 2). By doing this, if the image contains numerous edges, we still are able to correctly determine if a circular shape is included. However, the main problem with this approach is that an image is discrete; thus, the edge directions will include an error. Moreover, the circular shapes appearing on the image rarely correspond to a circle. This is particularly troublesome when trying to detect circular shapes with a "big" radius.

Our proposal still uses the edge directions. However, the difference is that it uses them to calculate an *error*, ϵ, between the ideal edge direction α and the real edge direction $\widehat{\alpha}$ as follows:

$$\epsilon\left(\alpha,\widehat{\alpha}\right)=\begin{cases}\left(\left|\alpha-\widehat{\alpha}\right|\right)/\pi & \text{if } \left|\alpha-\widehat{\alpha}\right|\leq\pi\\ \left(2\pi-\left|\alpha-\widehat{\alpha}\right|\right)/\pi & \text{otherwise}\end{cases}\;. \tag{1}$$

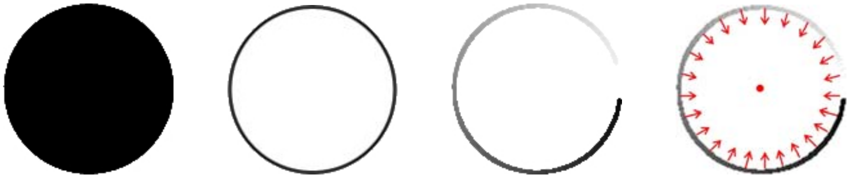


Fig. 2. (From left to right) 1) A circle. 2) The edges detected. 3) Degree of those edges (lighter elements are close to 0° while darker ones are closer to 360°). 4) By increasing the accumulator along the edge direction, the center of the circle is detected.

From this equation, the CHT is modified into the function `CDHT (edges, angles, r, ang_dif)`, and by adding between steps 4 and 5 the instruction:

```
4'.     $\widehat{\alpha}$ ← angles [i]
```

where `angles [i]` holds the angle of `edges [i]`. Also, we substituted instruction in step 10 with:

```
10.          acum [a, b] ← acum [a, b] + 1 - $\epsilon$ ($\alpha$, $\widehat{\alpha}$ )
```

Different methods can be applied to obtain from an image its edges and their angles. For this proposal, the generalization of Sobel [12] was used. To define it, let $p, q \in \mathbb{Z}^{n+1}$, with:

$$p_i = c_{n,i-1}, \tag{2}$$

$$q_i = \begin{cases} c_{n,i} - c_{n,i-1} & \text{if } i < n/2 \\ -c_{n,i} + c_{n,i-1} & \text{if } i > n/2 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $c_{n,k} := k!/\left(n!\left(n-k\right)!\right)$. From these two vectors, we obtain the template $T = pq^t$. T is used to calculate the magnitude of the edges of a grayscale image I in the x and y direction, denoted by M_x, M_y respectively, as:

$$M_x = I * T, M_y = I * T^t, \tag{4}$$

with $*$ denoting the $2D$ correlation. From M_x and M_y, the magnitude M is calculated as:

$$M = \sqrt{M_x^2 + M_y^2}, \tag{5}$$

while the direction of every pair $\left((M_x)_{i,j}, (M_y)_{i,j}\right)$ refers to the angle of this vector. From the magnitude, we define a edge as those pixels for which the magnitude is greater or equal to a threshold.

3 Experimental Results

To test the system, *CDHT* was implemented in MATLAB®. The implementation was used over three problems: 1) autocalibration, 2) detection in the presence of noise, and 3) detection of circular shapes with large unknown radii.

For the experiments, b denotes the edges coordinates of the input grayscale image, θ its associated directions, and the angle difference was set to one. From each image used for these experiments, the center coordinates $\widetilde{x}$ of those circular shapes appearing on it, as well as its radius $\widetilde{r}$, were experimentally determined. The coordinates of the circles detected by using *CDHT* , x^*, were compared by using the Euclidean norm over the difference $\Delta x := \widetilde{x} - x^*$, whilst for the radius, r^*, the absolute value of $\Delta r := \widetilde{r} - r^*$ was calculated.

3.1 Autocalibration

The autocalibration test consisted in presenting the system an image that contains a circular shape whose diameter is unknown. In this test, the system applies the *CDHT* for a number of radii and selects the one for which the accumulator holds the greatest value; *i.e.*:

$$r^* = \arg\max \left\{ \max \left\{ CDHT\left(b, \theta, r, 1\right) \right\} : r \in \left\{ r_{\min}, \ldots, r_{\max} \right\} \right\}. \tag{6}$$

In this particular case, $r_{\min} = 25, r_{\max} = 35$. On the other hand, x^* refers to the coordinates that hold the maximum value of $CDHT\left(b, \theta, r^*, 1\right)$ (the same was done for *CHT*).

The test was done over a set of sixty images of 320 by 240 from an eye captured in two different sessions with a web cam. The person was allowed to look into different directions. Because of this, the set was divided in two subsets. One containing 30 images for which the iris was about the middle of the eye[3]. The second subset contained the rest of the images[4]. For this subset, the radii difference was not calculated. The results are summarized in the next table.

Table 1. Results for the autocalibration test obtained through *CHT* and *CDHT*

	looking towards		looking away
	$mean\left(\lVert\Delta x\rVert_2\right)$	$mean\left(\lvert\Delta r\rvert\right)$	$mean\left(\lVert\Delta x\rVert_2\right)$
CDHT	0.8041	1.2667	1.6701
CHT	0.8834	1.0333	1.6121

Figure 3 shows some of the results attained during this test for *CDHT*.

3.2 Detection in the Presence of Noise

This test used the centered iris images subset. It is assumed that the radius r^* is already known. However, different kinds of noise (*Gaussian*, *Salt n' Pepper* and *Poisson*) were added to the images before determining the edges and their

[3] In these images the circular shape was closer to a circle

[4] These images contained circular shapes better described by an ellipse with greater eccentricities of those in the first set. This also explains the magnitued of the Δr

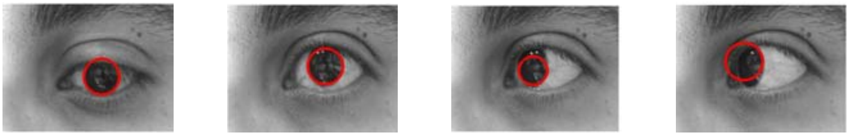


Fig. 3. Examples of the results obtained from the autocalibration test

magnitudes. Once this was done, the coordinates of $CDHT\,(b,\theta,r^{*},1)$ that hold its maximum value, x^{*}, are compared with the ideal, $\widetilde{x}$. If the euclidean norm of the difference Δx is less than or equal to $\sqrt{4}$, then the detection is qualified as *valid*, and *not valid* otherwise. For the process a Median Filter over a 5×5 window was used. The results of $CDHT$ and CHT are included in Table 2.

Table 2. Results for the circular shape detection in the presence of different kinds of noise

	Gauss,	*Gauss,*	*Salt n' Pepper*	*Poisson*
	$\mu = 0, \sigma = 0.05$	$\mu = 0, \sigma = 0.1$		$density = 0.1$
$CDHT$	83.33%	53.33%	100%	100%
CHT	100%	76.67%	100%	96.67%

3.3 Detection of Circular Shapes with Large Unknown Radii

One of the major problems of the CHT is the detection of circular shapes for a large radius. This problem follows from two major reasons: for large radii, the probability that the circular shape is perfect reduces considerably; the second reason, is that the probability to account a edge not related to a circular shape increases as a function of the radius. Consequently, the variations of the CHT that use the threshold and the direction of the edges for the detection seem like a natural alternative. However, since the edge direction variations "trace a line" in the direction of the edge detected, the accumulator rarely holds sufficient evidence of the circular shape for large radii. Clearly, this proposal is not affected by this matter, as it will increase the accumulator in a way proportional to how close the edge direction is to that expected.

To show this, a set of thirty satellite images of 256 by 256 from Mars containing craters of different sizes along with a number of other geographical features were used. For the test, no *a priori* information about the radii is assumed other than the range $r = \{15, 16, \ldots, 70\}$. The results of $CHT\,(b,\theta,r,1)$ and $CDHT\,(b,\theta,r,1)$ are compared to a threshold $th \in [0, 360]$. If the accumulator is greater or equal to th, then it is assumed that there is a circular shape for that particular location. Two selected elements were considered equivalent if the absolute difference between their center coordinates and their radius was for every element less than or equal to one. From this relation, the transitive closure was obtained, and the elements were clustered using this equivalence relationship. This test produces two types of error: a False Rejection (FR) and a False Acceptance (FA)[5]. The selected values were compared with those of the ideals and were qualified as *valid* if the euclidean norm of the difference Δx is less than or equal to $\sqrt{4}$ and the absolute difference of Δr is less than or equal to 3. The selected values not fulfilling these requirements were qualified as a FR or a FA. For the detection, the edges deletion of selected circular shapes was done as proposed in [13].

Table 3. Results of the Large Unknown Radii Test. The cells contain the Valid Rate, FA Rate, FR Rate. If either, the FA or FR Rate are greater than 0.5, or the Valid Rate is less than 0.5 then the data is omitted.

th	360×0.2	360×0.25	360×0.3
$CDHT$	(92.56%, 32, 27%, 11.37%)	(85.83%, 18.93%, 17.46%)	(74.91%, 31.56%, 28.65%)
CHT	–	(81.25%, 43.48%, 27.43%)	(71.88%, 22.58%, 31.85%)

Some results are included in Figure 4.

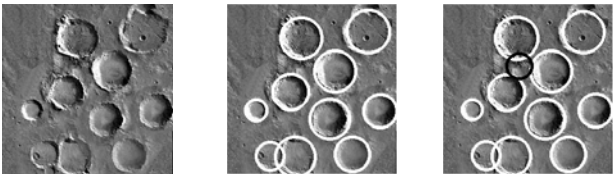


Fig. 4. Examples of the results obtained from the large unknown radii test. From left to right, original image, results obtained with the best threshold for CDHT and CHT. For the second and third image black and white circles represent FA and valid results respectively.

4 Conclusions

During this paper, we presented the CDHT. This variation of CHT calculates the data of the circular shapes appearing within an image with the same increasing the algorithm complexity, which is $\Theta(m)$, with $m = |\texttt{edges}|$.

For the Autocalibration Test, the results using both techniques were equivalent. This was to be expected, since only one circular shape in contained within every image, and it is clearly contrasted from the rest of the image.

For the Detection in the Presence of Noise Test, we also expected to obtain similar results for both techniques. However, CHT proved to be better for the Gaussian noise. The median filter was used since it is known that it preserves the edges of the filtered image. However, the square template altered the direction of the edges related to the iris. Thus, the recognition was not as good as desired.

[5] FA: if an existing crater is not detected. FR: If a not existing crater is detected.

The best performance of CDHT was obtained in the Large Unknown Radii Test. This is important, since this is the hardest test or the three proposed in this paper. Even more, this is probably the most likely scenario for practical applications. In this case, CDHT obtained the best ratios VR:FR and FR:FA.

References

1. Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E.: Circle detection on images using genetic algorithms. Pattern Recognition Letters 27(6), 652–657 (2006)
2. Hough, P.V.C.: Methods and Means for Recognizing Complex Patterns. U.S. Patent 3, 069, 654 (1962)
3. Duda, R.O., Hart, P.E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. Comm. ACM 15, 11–15 (1972)
4. Kimme, C., Ballard, D., Sklansky, J.: Finding Circles by an Array of Accumulators. Communications of the ACM 18(2), 120–122 (1975)
5. Tsuji, S., Matsumoto, F.: Detection of ellipses by a modified Hough transformation. IEEE Transactions on Computers C-27(8), 777–781 (1978)
6. Xu, L., Oja, E., Kultanan, P.: A new curve detection method: randomized Hough transform (RHT). Pattern Recognition Letter 11(5), 331–338 (1990)
7. Aguado, A.S., Montiel, E., Nixon, M.S.: On Using Directional Information for Parameter Space Decomposition in Ellipse Detection. Pattern Recognition 28(3), 369–381 (1996)
8. McLaughlin, R., Alder, M.: The Hough transform versus UpWrite. IEEE Trans, PAMI 20(4), 396–400 (1998)
9. Salamuniccar, G., Loncaric, S.: Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on MOLA data. Advances in Space Research 42(1), 6–19 (2008)
10. Benn, D.E., Nixon, M.S., Carter, J.N.: Robust Eye Centre Extraction Using the Hough Transform. In: Bigün, J., Borgefors, G., Chollet, G. (eds.) AVBPA 1997. LNCS, vol. 1206, pp. 3–9. Springer, Heidelberg (1997)
11. Kaminka, G., Lima, P., Rojas, R. (eds.): RoboCup 2002. LNCS (LNAI), vol. 2752. Springer, Heidelberg (2003)
12. Nixon, M.S., Aguado, A.S.: Feature Extraction and Image Processing, 2nd edn. Academic Press, London (2007)
13. Flores-Méndez, A.: Crater Marking and Classification Using Computer Vision. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 79–86. Springer, Heidelberg (2003)

VI Analysis of Signal, Speech and Language

Isolate Speech Recognition Based on Time-Frequency Analysis Methods

Alfredo Mantilla-Caeiros[1], Mariko Nakano Miyatake[2], and Hector Perez-Meana[2]

[1] Intituto Tecnologico de Monterrey, Campus Ciudad de Mexco,
Av. Del Puente Mexico D.F.
[2] ESIME Culhuacan, Instituto Politécnico Nacional,
Av. Santa Ana 1000, 04430 Mexico D.F. Mexico
amantill@itesm.mx, mariko@infinitum.com.mx,
hmpm@prodigy.net.mx

Abstract. A feature extraction method for isolate speech recognition is proposed, which is based on a time frequency analysis using a critical band concept similar to that performed in the inner ear model; which emulates the inner ear behavior by performing signal decomposition, similar to carried out by the basilar membrane. Evaluation results show that the proposed method performs better than other previously proposed feature extraction methods when it is used to characterize normal as well as esophageal speech signal.

Keywords: Feature extraction, inner ear model; isolate speech recognition, time-frequency analysis.

1 Introduction

The performance of any speech recognition algorithm strongly depends on the accuracy of the feature extraction method, because of that several methods have been proposed in the literature to estimate a set of parameters that allows a robust characterization of the speech signal. A widely used feature extraction method consists on applying the Fast Fourier Transform (FFT) to the speech segment under analysis. This representation in the frequency domain is obtained by using the well-known MEL scale, where the frequencies smaller than 1kHz are analyzed using a linear scale, while the frequencies larger than 1kHz are analyzed using a logarithmic scale, with the purpose of creating an analogy with the internal cochlea of the ear that works as a frequencies splitter [1]-[4].

Linear Predictive Coding (LPC) is other widely used feature extraction method whose purpose is to find set of parameters that allows an accurate representation of the speech signal as the output of an all pole digital filter, which models the vocal track, whose excitation is an impulse sequence with a period equal to the pitch period of speech signal under analysis, when the speech segment is a voiced one, or a white noise when the speech segment is an unvoiced one [1], [3]. Here, to estimate the features vector, firstly the speech signal is divided in segments of 20 to 25 ms, with 50% of overlap. Finally, the linear predictive coefficients of each segment are

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 297–304, 2009.

estimated such that the mean square value of prediction error becomes a minimum. Because five formants or resonant frequencies are enough to characterize the vocal track, a predictive filter of order 10 is enough [1], [4]. Depending on the application, it may be useful to take the LPC average of the N segments contained in the word under analysis, such that this coefficients average may be used as the behavior model of a given word. Thus the averaged m-th LPC becomes

$$\hat{a}_m = \frac{1}{N}\sum_{i=1}^{N} a_{i,m}, \quad 1 \le m \le p \tag{1}$$

where N is the total number of segments contained in the word.

The cepstral coefficients estimation is other widely used feature extraction method in speech recognition problems. These coefficients form a very good features vector for the development of speech recognition algorithms [1, 2, 4], sometimes better than the LPC ones. The cepstral coefficients can be estimated from the LPC coefficients applying the following expression [1]

$$c_n = -a_n - \frac{1}{n}\sum_{i=1}^{n-1} (n-i) a_i c_{n-i} \tag{2}$$

where C_n is the n-th LPC-Cepstral coefficients, a_i is the i-th LPC coefficients and n is the Cepstral index. Usually the number of cepstral coefficients is equal to the number of LPC ones to avoid noise [1]. For isolated word recognition, it is possible to take also the average of cepstral coefficients contained in the word to generate an averaged feature vector (CLPC) to be used during the training or during the recognition task. Most widely used feature extraction methods, such as those describe above, are based on modeling the vocal tract. However if the speech signals are processed taking in account the form in which they are perceived by the human ear, similar or even better results may be obtained. Thus in [5] the use of time-frequency analysis and auditory modeling is proposed, in reference [6] an automatic speech recognition scheme using perceptual features is proposed. Thus to use an ear model-based feature extraction method may be an attractive alternative because, this approach allows characterizing the speech signal in the form that it is perceived [7].

This paper proposes a feature extraction method for speech recognition, based on an inner ear model that takes in account the fundamentals concepts of critical bands. Evaluation results using normal and esophageal speech show that the proposed approach provides better results than other previously feature extraction methods.

2 Feature Extraction Based on Inner Ear Model

In the inner ear, the basilar membrane carries out a time-frequency decomposition of the audible signal through a multi-resolution analysis similar to that performed by a wavelet transform [6]. Thus to develop a feature extraction method that emulates the basilar membrane operation, it must be able to carry out a similar decomposition, as proposed in the inner ear model developed by Zhang et. al. [8]. In this model the dynamics of basilar membrane, which has a characteristic frequency equal to f_c, can be modeled by a gamma distribution multiplied by a pure tone of frequency f_c, that is using the so-called gamma-tone filter. Here the shape of the gamma distribution is related to the filter order, while the scale is related to the inverse of the frequency of occurrence of events under analysis, when they have a Poisson distribution. Thus the gamma-tone filter representing the impulse response of the basilar membrane is given by [8]

$$\psi_\theta^\alpha(t)=\frac{1}{(\alpha-1)!\,\theta^\alpha}t^{\alpha-1}e^{\frac{-t}{\theta}}\cos(2\pi t/\theta)\quad t>0 \tag{3}$$

where α and θ are the shape and scale parameters, respectively. Equation (3) defines a family of gamma-tone filters characterized by θ and α, thus it is necessary to look for the more suitable filter bank to emulate the basilar membrane behavior. To this end, we can normalize the characteristic frequency by setting $\theta=1$ and $\alpha=3$, which according to the basilar membrane model given by Zhang et al [8], provides a fairly good approximation to the inner ear dynamics. Thus from (3) we get

$$\psi(t)=\frac{1}{2}t^2e^{-t}\cos(2\pi t)\quad t>0 \tag{4}$$

This function presents the expected attributes of a mother wavelet because it satisfies the admissibility condition given by [9], [10]

$$\int_{-\infty}^{\infty}|\psi(t)|^2\,dt=\frac{1}{2}\int_0^{\infty}\left|t^2e^{-t}\cos(2\pi t\right|^2dt<\infty \tag{5}$$

That means that the norm of $\psi(t)$ in $L^2(\mathbf{R})$ space exists and then the functions given by (4) constitutes an unconditional basis for $L^2(\mathbf{R})$. This fact can be proven by using the fact that [11]

$$\int_0^{\infty}\Psi(s\omega)\frac{ds}{s}<\infty \tag{6}$$

The previous statement can verified substituting the Fourier transform of (4), Ψ(ω), into (6), where

$$\Psi(\omega)=\frac{1}{2}\left[\frac{1}{[1+j(\omega-2\pi)]^2}+\frac{1}{[1+j(\omega+2\pi)]^2}\right] \tag{7}$$

Thus we can generate the expansion coefficients of an audio signal *f(t)* by using the scalar product between *f(t)* and the function *ψ(t)* with translation τ and scaling factor s as follows [11]

$$\gamma(\tau, s) = \frac{1}{\sqrt{s}} \int_{0}^{\infty} f(t)\psi\left(\frac{t-\tau}{s}\right)dt \tag{8}$$

A sampled version of (8) must be specified because we require recognizing discrete time speech signals. To this end, a sampling of the scale parameter, s, involving the psychoacoustical phenomenon known as critical bandwidths will be used [10].

The critical bands theory models the basilar membrane operation as a filter bank in which the bandwidth of each filter increases as its central frequency increases [8, 9]. This statement allows defining the Bark frequency scale; a logarithmic scale in which the frequency resolution of any section of the basilar membrane is exactly equal one Bark, regardless of its characteristic frequency. Because the Bark scale is characterized by a biological parameter, there is not an exact expression for it, given as a result several different proposals available in the literature. Among them, the statistical fitting provided by Schroeder et al [10], appears to be a suitable choice. Thus using the approach provided by [8], the relation between the linear frequency, *f*, given in Hz and the Bark frequency, *Z*, is given by [10]

$$Z = 7\ln\left(\frac{f}{650} + \sqrt{\left(\frac{f}{650}\right)^2 + 1}\right) \tag{9}$$

Next by using the expression given by (9), the central frequency in Hz corresponding to each band in the Bark frequency scale becomes [10]

$$f_c = 325 \cdot \frac{e^{\frac{2j}{7}} - 1}{e^{\frac{j}{7}}} \quad j = 1,2,\ldots \tag{10}$$

Next, using the central frequencies given by (10) the jth scale factor is given by

$$s_j = \frac{1}{f_c} = \frac{1}{325} \cdot \frac{e^{\frac{j}{7}}}{e^{\frac{2j}{7}} - 1} \quad j = 1,2,\ldots \tag{11}$$

The inclusion of bark frequency in the estimation of scaling factor, as well as the relation between (4) and the dynamics of basilar membrane, allows frequency decomposition similar to that carried out in the human hearing. The scaling factor give by (11) satisfies the Littlewood-Paley theorem since

$$\lim_{j\to+\infty} \frac{s_{j+1}}{s_j} = \lim_{j\to+\infty} \frac{e^{(j+1)/7}\left(e^{2j/7} - 1\right)}{e^{j/7}\left(e^{2(j+1)/7} - 1\right)} = \lim_{j\to+\infty} \frac{e^{(3j+1)/7}}{e^{(3j+2)/7}} = e^{-1/7} \neq 1 \tag{12}$$

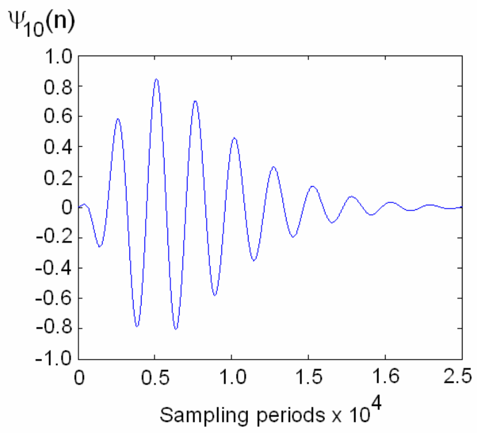


Fig. 1. 10th Gammatone function derived from the inner ear model

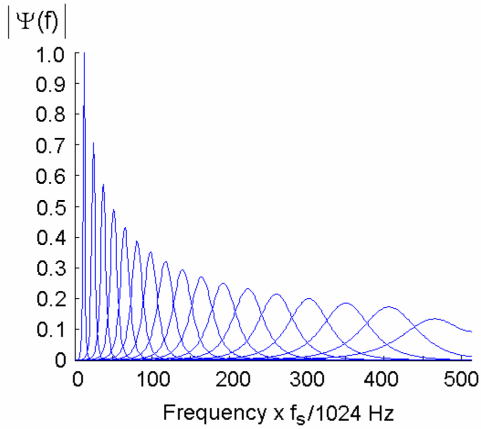


Fig. 2. Frequency response of filter bank derived from an inner ear model

Then there is not information loss during the discretization process. Finally the number of subbands is related with the sampling frequency as follows

$$j_{\max} = \operatorname{int}\left(7\ln\left(\frac{f_s}{1300} + \sqrt{\left(\frac{f_s}{1300}\right)^2 + 1}\right)\right) \tag{13}$$

Thus for a sampling frequency equal to 8KHz the number of subbands becomes 17. Finally, the translation axis is naturally sampled because the input data is a discrete time signal, and then the expansion coefficients can be estimated as follows [9]

$$C_{f,\psi}(\tau)=\sum_{-\infty}^{\infty} f(n)\psi_s(n-\tau) \tag{14}$$

where

$$\psi_s(n)=\frac{1}{2}(nT/s)^2 e^{-(nT/s)}\cos(2\pi nT/s))\ n>0 \tag{15}$$

where T denotes the sampling period. Here the expansion coefficients $C_{f,\psi}$ obtained for each subband are used to carry out the recognition task. Figures 1 shows $\psi_{10}(n)$, and Fig. 2 shows the filter bank power spectral density, respectively.

3 Evaluation Results

The performance of proposed feature extraction method was evaluated in isolate word recognition tasks, with normal as well as esophageal speech signals. Here the feature vector consists of the following parameters: the *m-th* frame energy given by [12]

$$\bar{x}_m(n)=\gamma\bar{x}_m(n-1)+x_m^2(n),\ \ n=1,2,..,N\ , \tag{16}$$

where (N=1/γ), the energy contained in each one of the 17 wavelet decomposition levels,

$$\bar{y}_{k,m}(n)=\gamma\bar{y}_{k,m}(n-1)+y_{k,m}^2(n),\quad k=1,2,...,17\ , \tag{17}$$

the difference between the energy of the previous and actual frames,

$$v_0(m)=\bar{x}_m(N)-\bar{x}_m(N-1)\ , \tag{18}$$

together with the difference between the energy contained in each one of the 17 wavelet decomposition levels of current and previous frames,

$$v_k(m)=\bar{y}_k(m)-\bar{y}_k(m-1),\quad k=1,2,...,17\ . \tag{19}$$

where m is the number frame. Then the feature vector derived using the proposed approach becomes

$$\mathbf{X}(m)=\left[\bar{x}_m(N),\bar{y}_{1,m}(N),..,\bar{y}_{17,m}(N),\bar{v}_0(m),\bar{v}_1(m),..,\bar{v}_{17}(m)\right] \tag{20}$$

Here the last eighteen members of the feature vector include the spectral dynamics of speech signal concatenating the variation from the past feature vector to the current one.

To evaluate the actual performance of proposed approach it was compared with the performance provided by others conventional methods like Mel Frequency Cepstral

Coefficients (MFCC), Linear Prediction Coefficients (LPC), Dubechies wavelet function [9] and Haar transform [9] when they are required to perform isolate work recognition tasks, using a data base developed with the assistance of Institute of Human Communication of The National Rehabilitation Institute of Mexico: The data base was developed using a 1.7GHz DELL Inspiron 8200 Pentium 4-M , with a Sony F-V220 Dynamic Microphone and an audio board Crystal WDM Audio from Cirrus Logic Inc. The data base consists of 100 words of 20 normal speakers and 20 esophageal speakers. Evaluation results provided in Table 1 shows that proposed approach provides better recognition performance than other widely used feature extraction methods. In all cases the feature vectors were estimated in similar form, with 100 words of 20 different speakers, and used as input of a recursive neural network [11]. Here half words were used for training and half for testing. Finally table 2 shows the performance of proposed approach when is used with two different pattern classification methods, the neural network and hidden Markov Model.

Table 1. Comparison between several features extraction methods using normal and esophageal speeker voice

	Proposed	**Daub 4**	**Haar**	**LPC**	**MFCC**
Normal Speaker	**97%**	**83%**	**70%**	**94%**	**95%**
Esophageal Speaker	**93%**	**77%**	**52%**	**89%**	**90%**

Table 2. Recognition performance of proposed feature extraction method when is used with two different identification algorithms

Classifier	**Normal speech**	**Esophageal speech**
Recurrent Neural Network	**95%**	**93%**
Hidden Markov Models	**92%**	**92%**

Evaluation results show that proposed algorithm performs better than other previously proposed feature extraction methods, when it is used to recognize isolated normal speech, as well as isolated esophageal speech signals.

4 Conclusions

A new feature extraction based on an inner ear model was proposed, and applied to feature extraction in isolate word recognition for normal and esophageal speech. The evaluation results performed using real speech data show that the proposed approach, based on modeling the basilar membrane, accurately extracts perceptually meaningful data required in isolate word recognition; providing better results than others feature extraction methods. The use of artificial neural network as a classifier produced success rate higher than 97% in the recognition of Spanish word pronounced by normal speakers and 93% when the words are pronounced by esophageal speaker. An important consequence from the use of multi-resolution analysis techniques is that high frequency information is captured during the feature extraction stage.

Acknowledgements

The authors thank the National Science and Technology Council for the financial support during the realization of this research; and to Dr. Xochiquetzal Hernandez from The National Rehabilitation Institute of Mexico for the assistance provided to develop the speech database used to evaluate the proposed method.

References

1. Rabiner, L., Juang, B.: Fundamentals of Speech Recognition. Prentice Hall, Piscataway (1993)
2. Rabiner, R., Juang, B.H., Lee, C.H.: An Overview of Automatic Speech Recognition. In: Lee, C.H., Soong, F.K., Paliwal, K.K. (eds.) Automatic Speech and Speaker Recognition: Advanced Topics, pp. 1–30. Kluwer Academic Publisher, Dordrecht (1996)
3. Junqua, C., Haton, J.P.: Robustness in Automatic Speech Recognition. Kluwer Academic Publishers, Dordrecht (1996)
4. Pitton, J.W., Wang, K., Juang, B.H.: Time-frequency analysis and auditory modeling for automatic recognition od speech. Proc. of The IEEE 84(9), 1109–1215 (1999)
5. Haque, S., Togneri, R., Zaknich, A.: Perceptual features for automatic speech recognition in noise environments. Speech Communication 51(1), 58–75 (2009)
6. Suarez-Guerra, S., Oropeza-Rodriguez, J.: Introduction to Speech Recognition. In: Perez-Meana, H. (ed.) Advances in Audio and Speech Signal Processing; Technologies and Applications, pp. 325–347. Idea Group Publishing, USA (2007)
7. Childers, D.G.: Speech Processing and Synthesis Toolboxes. Wiley and Sons, New York (2000)
8. Zhang, X., Heinz, M., Bruce, I., Carney, L.: A phenomenological model for the responses of auditory-nerve fibers: I. Nonlinear tuning with compression and suppression. Acoustical Society of America 109(2), 648–670 (2001)
9. Rao, R.M., Bopardikar, A.S.: Wavelets Transforms, Introduction to Theory and Applications. Addison Wesley, New York (1998)
10. Schroeder, M.R., et al.: Objective measure of certain speech signal degradations based on masking properties of the human auditory perception. In: Frontiers of Speech Communication Research. Academic Press, London (1979)
11. Freeman, J., et al.: Neural Networks, Algorithms, Applications and Programming Techniques. Addison-Wesley, New York (1991)
12. Mantilla-Caeiros, A., Nakano-Miyatake, M., Perez-Meana, H.: A New Wavelet Function for Audio and Speech Processing. In: Proc. of the MWSCAS 2007, pp. 101–104 (2007)

Feature Selection Based on Information Theory for Speaker Verification

Rafael Fernández[1,2], Jean-François Bonastre[2], Driss Matrouf[2], and José R. Calvo[1]

[1] Advanced Technologies Application Center, Havana, Cuba
[2] Laboratoire d'Informatique d'Avignon, UAPV, France
{rfernandez,jcalvo}@cenatav.co.cu
{jean-francois.bonastre,driss.matrouf}@univ-avignon.fr

Abstract. Feature extraction/selection is an important stage in every speaker recognition system. Dimension reduction plays a mayor roll due to not only the *curse* of dimensionality or computation time, but also because of the discriminative relevancy of each feature. The use of automatic methods able to reduce the dimension of the feature space without losing performance is one important problem nowadays. In this sense, a method based on mutual information is studied in order to keep as much discriminative information as possible and the less amount of redundant information. The system performance as a function of the number of retained features is studied.

Keywords: mutual information, feature selection, speaker verification.

1 Introduction

The task of feature extraction/selection is a crucial step in an automatic speaker recognition system. The performance of the later components –speaker modeling and pattern matching– is strongly determined by the quality of the features extracted in this first stage [1,2]. Most of the efforts today are doveted to the classification stage, and little to find optimal representations of the speakers.

Methods like Principal Component Analysis (PCA), Discrete Cosine Transform (DCT) and Linear Discriminant Analysis (LDA) have been employed extensively in the literature for reducing the dimension of the feature space. These methods rely on maximazing the data distribution variance –globally or per class– or minimizing the reconstruction error by selecting only a subset of the original feature set. However, this does not mean necessarily high speaker discrimination or low redundancy in the feature set. Others like the *knock-out* method [3,4] proposes to evaluate all the subsets of $n-1$ coefficients in order to keep the best subset in each iteration. One problem here is that the selection of the best subset strongly depends on the classification method employed.

Various combinations of LFCC-based features, the energy, and their deltas and delta-deltas were tested in [5] to obtain the best configuration. In this work,

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 305–312, 2009.

we study the use of an information theory based method [6] in order to select automatically the best subset of acoustic coefficients. This method was applied in [7] for selecting the best Wavelet Packet Tree (WPT) in a speaker identification system. The main principle of the selection algorithm is to maximize the discriminative information while minimizing the redundancy between the selected features.

A study of the system performance as a function of the dimension of the feature space is presented. A state-of-art speaker verification system [8] is used in order to evaluate the usefulness of the method.

The remainder is organized as follows: Section 2 shows the basis of the proposed information theory oriented method; Section 3 shows the details of the implemented algorithm; database description, experimental work and results are summarized in Section 4; last section is devoted to conclusions and future works.

2 Feature Selection Based on Mutual Information

Reducing the dimensionality of feature vectors is usually an essential step in pattern recognition tasks. By removing most irrelevant and redundant features, feature selection helps to improve the performance of learning models by: alleviating the effect of the *curse* of dimensionality, enhancing generalization capability, speeding up learning process and improving model interpretability.

Methods based on Information Theory can act as a general criterion, since they consider high order statistics, and can be used as a base for nonlinear transformations [9]. With these methods, low information redundancy is achieved and the discriminative information is intended to be kept while reducing the dimensionality.

In probability theory and information theory, the mutual information of two random variables is a quantity that measures their mutual dependence [10].

Let $\mathcal{S}$ and $\boldsymbol{X} \in \mathbb{R}^N$ be the variables for the speaker class and the speech feature vector respectively. The mutual information between $\mathcal{S}$ and $\boldsymbol{X}$ is given by:

$$I(\mathcal{S}, \boldsymbol{X}) = H(\mathcal{S}) + H(\boldsymbol{X}) - H(\mathcal{S}, \boldsymbol{X}) = H(\mathcal{S}) - H(\mathcal{S}|\boldsymbol{X}), \tag{1}$$

where $H(\cdot)$ is the entropy function, which is a measure of the uncertainty of the variable. For a discrete-valued random variable $\boldsymbol{X}$, it is defined as:

$$H(\boldsymbol{X}) = -\sum_{m} p(\boldsymbol{X} = \boldsymbol{x}_m) \log_2 p(\boldsymbol{X} = \boldsymbol{x}_m), \tag{2}$$

where $p(\boldsymbol{X} = \boldsymbol{x}_m)$ is the probability that $\boldsymbol{X}$ takes the value $\boldsymbol{x}_m$.

From (1), mutual information measures the uncertainty reduction of $\mathcal{S}$ knowing the feature values. Those features with low speaker information have low values of mutual information with the speaker class. Following this criterion, the best K coefficients from the original set $\boldsymbol{X} = \{X_1, \ldots, X_N\}$ are those $\boldsymbol{X}' = \{X_{i_1}, \ldots, X_{i_K}\} \subset \boldsymbol{X}$ which maximise the mutual information with the speaker class:

$$\boldsymbol{X}' = \underset{\{X_{j_1}, \ldots, X_{j_K}\}}{\arg\max}\; I(\mathcal{S}, \{X_{j_1}, \ldots, X_{j_K}\}). \tag{3}$$

If the features were statistically independent, the search in (3) would be reduced to find those features iteratively. If we know the first $k-1$ features, the k-th is obtained using this recursive equation:

$$X_{i_k} = \underset{X_j \notin \{X_{i_1},\ldots,X_{i_{k-1}}\}}{\arg\max} I(X_j, \mathcal{S}), \qquad k = 1, \ldots, K. \tag{4}$$

In the case of statistically dependent feature –very frequent in real life problems– the latter is not true. Here, the problem of finding out the best subset (see Eq. (3)) becomes a search for all the $\binom{N}{K}$ combinations.

In order to select the best coefficients, the sub-optimal method [6,11] was applied. If we have the first $k-1$ coefficients $\boldsymbol{X}_{k-1} = \{X_{i_1}, \ldots, X_{i_{k-1}}\}$, the k-th is selected according to:

$$X_{i_k} = \underset{X_j \notin \boldsymbol{X}_{k-1}}{\arg\max} \left[I(X_j, \mathcal{S}) - \frac{1}{k-1} \sum_{s=1}^{k-1} I(X_j, X_{i_s}) \right]. \tag{5}$$

The idea is to look for the coefficients with high mutual information with the speaker class and low average mutual information with the features previously selected. Last term in (5) can be thought of as a way to reduce the redundant information. Here, mutual information between two variables is the only estimation needed, which avoids the problem of estimating the probability densities of high dimension vectors. We used histogram method to calculate the probability densities.

3 The Algorithm

Based on the stated above, we developed the following algorithm to withdraw the worst features in an original 60-feature LFCC configuration.

Algorithm 1. Proposed method

```
$k = N$;
$SearchList = \{1, \ldots, N\}$;
while $k > K$ do
    foreach $n \in SearchList$ do
        $C = SearchList \setminus \{n\}$;
        $F(n) = I(X_n, \mathcal{S}) - \frac{1}{k-1} \sum_{m \in C} I(X_n, X_m)$;
    end
    $n^* = \arg\min_m (F(m))$;
    $SearchList = SearchList \setminus \{n^*\}$;
    $k = k + 1$;
end
```

At every stage, the coefficient to be eliminated is selected according to (5). This cycle is repeated until the desired number of K features is reached.

4 Experiments and Results

4.1 Database

All the experiments presented in section 4 are performed based upon the NIST 2005 database, 1conv-4w 1conv-4w, restricted to male speakers only. This condition consists of 274 speakers. Train and test utterances contain 2.5 minutes of speech on average (extracted from telephone conversations). The whole speaker detection experiment consists in 13624 tests, including 1231 target tests and 12393 impostors trials. From 1 to 170 tests are computed by speaker, with an average of 51 tests.

4.2 Front End Processing

All the experiments were realized under the LIA_SpkDet system [12] developed at the LIA lab. This system consists in a cepstral GMM-UBM system and has been built from the ALIZE platform [8]. Depending on the starting set of coefficients, two cases –described below– were analyzed. The feature extraction is done using SPRO [13]. Energy-based frame removal –modelled by a 3 component GMM– is applied as well as mean and variance normalization. The UBM and target models contain 512 Gaussian components. LLR scores are computed using the top ten components. For the UBM, a set of 2453 male speakers from the NIST 2004 database was used.

The performance is evaluated through classical DET performance curve [14], Equal Error Rate (EER) and Detection Cost Function (DCF).

4.3 Experiments

Two experiments were done in order to select the best coefficients. In the first one a 60-feature set was taken as starting set, and in the second one, a 50-feature subset was considered.

Experiment 1: The starting set considered in this experiment consists in a 60-feature set composed of 19 LFCC, the energy and their corresponding first and second derivative.

Experiment 2: In this case the selection started from a 50-feature subset –derived from the previously described set– composed of the first 19 LFCC, their first derivative, the first 11 of the second derivative and the delta energy. This configuration is the result of large empirical experience based on human expert knowledge [5] and will be taken as the baseline in this work.

The results of the EER and DCF for each experiment are shown in Figures 1 and 2.

In order to analyze the eliminated features at each stage, the rank order for each coefficient for both starting sets is shown in figure 3. For better understanding they were divided in three classes: static, delta (D), and delta-delta (DD).

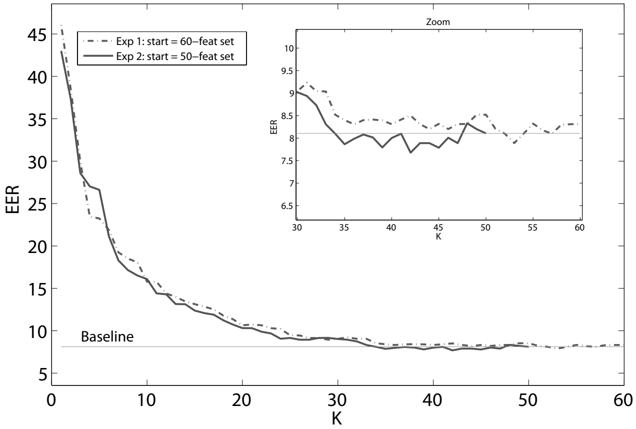


Fig. 1. EER as a function of the feature space dimension

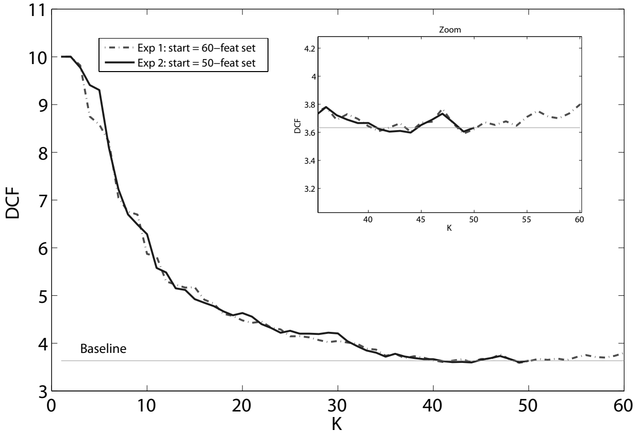


Fig. 2. DCF as a function of the feature space dimension

A general observed behavior is that even when all the features were used in the selection (Experiment 1), almost all the delta-delta features were the first eliminated. This is in accordance with the experience accumulated that state that these features have a weak contribution to the speaker verification task, since they do not carry a significant amount of new information. Meanwhile, by and large the static parameters show the highest relevance for both experiments. Static coefficient 'zero' was the least ranked among the statics coefficients as expected, although it was the best ranked among the delta-deltas.

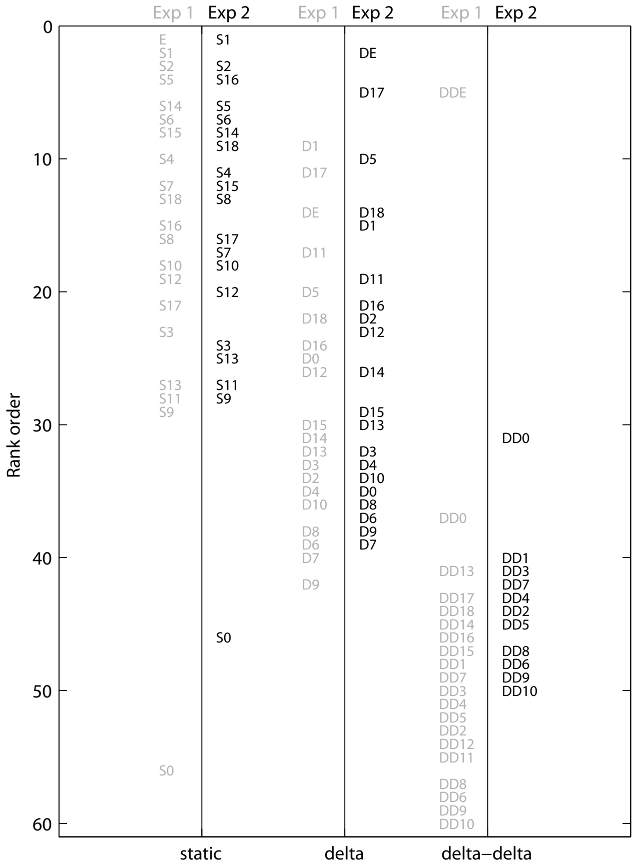


Fig. 3. Feature rank order for both experiments

One significant difference is the first place of the energy when all the features are considered, which was not used in the 50-feature starting set. However, when the selection starts from this configuration (Experiment 2), the system achieved the best results. This may be determined by all the *a priori* information that it includes, which is based in strong experimental basis and human expert knowledge. However, for both starting sets, the feature selection method was able to detect better configurations with a lower dimension.

The DET curves for some interesting configurations are shown in figure 4. Three configurations derived from the Experiment 2 are shown: the first one is the configuration with the smallest dimensionality that achieves at least the same performance as the baseline (K=34), the second one achieved the best performance among all the analyzed combinations (K=42), and the the third one is the baseline (K=50). The result for the full 60-feature set (K=60) is also presented.

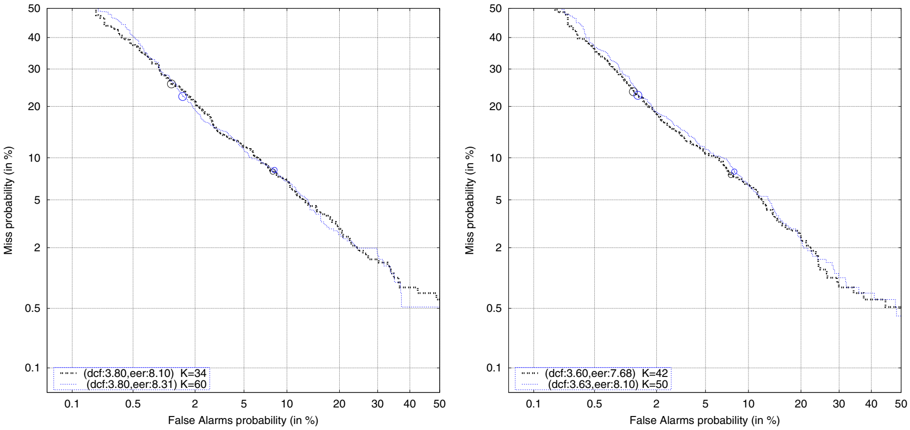


Fig. 4. DET curves for the configurations K=50 (baseline), K=34 and K=42, corresponding to the 50-feature set as a starting set for the selection

For almost all the operation points the configuration corresponding to K=42 outperforms the baseline (slightly, though). More significant is the configuration corresponding to K=34, which leads to the same EER as the baseline with a reduced number of features.

5 Conclusions

The problem of selecting the best LFCC subspace for speaker verification is discused in this work. A mutual information criterion has been studied to select the best LFCC features. The proposed feature selection method showed good capabilities for reducing the feature space dimension without losing performance. The experiment starting with *a priori* information finds better configurations. Better models must be studied in order to look for the optimal configuration with no *a priori* information. Other ways to find the most informative time-spectral regions will be analysed in the future.

References

1. Campbell, J.P.: Speaker recognition: A tutorial. Proceedings of the IEEE 85(9), 1437–1462 (1997)
2. Kinnunen, T.: Spectral features for automatic text-independent speaker recognition. Lic. Th., Department of Computer Science, University of Joensuu, Finland (2003)
3. Sambur, M.R.: Selection of acoustic features for speaker identification. IEEE Trans. Acoust. Speech, Signal Processing 23(2), 176–182 (1975)
4. Aha, D.W., Bankert, R.L.: A comparative evaluation of sequential feature selection algorithms. In: Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, pp. 1–7. Springer, Heidelberg (1995)

5. Fauve, B.: Tackling Variabilities in Speaker Verification with a Focus on Short Durations. PhD thesis, School of Engineering Swansea University (2009)
6. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancy. IEEE Trans. Patt. Anal. and Mach. Intel. 27(8), 1226–1238 (2005)
7. Fernández, R., Montalvo, A., Calvo, J.R., Hernández, G.: Selection of the best wavelet packet nodes based on mutual information for speaker identification. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 78–85. Springer, Heidelberg (2008)
8. Bonastre, J.F., et al.: ALIZE/spkdet: a state-of-the-art open source software for speaker recognition, Odyssey, Stellenbosch, South Africa (January 2008)
9. Torkkola, K., Campbell, W.M.: Mutual information in learning feature transformations. In: Proc. Int. Conf. on Mach. Learning, San Francisco, CA, USA, pp. 1015–1022. Morgan Kaufmann Publishers Inc., San Francisco (2000)
10. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, Hoboken (1991)
11. Lu, X., Dang, J.: Dimension reduction for speaker identification based on mutual information. In: Interspeech, pp. 2021–2024 (2007)
12. LIA_SpkDet system web site:
http://www.lia.univ-avignon.fr/heberges/ALIZE/LIA_RAL
13. Gravier, G.: SPRO: a free speech signal processing toolkit,
http://www.irisa.fr/metiss/guig/spro/
14. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assessment of detection task performance. In: Eurospeech, Rhodes, Greece, September 1997, pp. 1895–1898 (1997)

Simple Noise Robust Feature Vector Selection Method for Speaker Recognition

Gabriel Hernández, José R. Calvo, Flavio J. Reyes, and Rafael Fernández

Advanced Technologies Application Center
{gsierra,jcalvo,freyes,rfernandez}@cenatav.co.cu
http://www.cenatav.co.cu

Abstract. The effect of additive noise in a speaker recognition system is known to be a crucial problem in real life applications. In a speaker recognition system, if the test utterance is corrupted by any type of noise, the performance of the system notoriously degrades. The use of a feature vector selection to determine which speech frames are less affected by noise is the purpose in this work. The selection is implemented using the euclidean distance between the Mel features vectors. Results reflect better performance of robust speaker recognition based on selected feature vector, as opposed to unselected ones, in front of additive noise.

Keywords: speaker verification, cepstral features, selected feature vector, channel mismatch.

1 Introduction

Speech signal varies due to differences introduced by microphone, telephone, gender, age, and other factors, but a key problem is the presence of noise in the signal, which can provoke an awful performance in the speech processing algorithms working under extreme noisy conditions. Wireless communications, digital hearing aids or robust speech recognition, are examples of such systems which frequently require a noise reduction technique.

Recently, much research has been conducted in order to reduce the effect of handset/channel mismatch in speech and speaker recognition. Linear and nonlinear compensation techniques have been proposed, in the (a) feature, (b) model and (c) match-score domains [1]:

(a) Feature compensation methods [2]: filtering techniques such as cepstral mean subtraction or RASTA, discriminative feature design, and other feature transformation methods such as affine transformation, magnitude normalization, feature warping and short time Gaussianization.

(b) Model compensation methods [3]: speaker-independent variance transformation, speaker models transformation from multi-channel training data, and model adaptation methods.

(c) Score compensation methods [4]: aims to remove handset-dependent biases from the likelihood ratio scores as, H-norm, Z-norm, and T-norm.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 313–320, 2009.

Other methods to reduce specifically the impact of noise have been proposed [1]:

- filtering techniques,
- noise compensation,
- use of microphone arrays and,
- missing-feature approaches.

The features most commonly used are: static and dynamic Mel Frequency Cepstral Coefficients (MFCC), energy, zero crossing rate and pitch frequency. The classification methods commonly used are: Frame and utterance energy threshold, noise level tracking or model based. This paper investigates a feature vector selection method over MFCC in speaker recognition, using speech samples distorted by noise. This features vectors are selected by mean of clustering of the MFCC using as criterion of Euclidean distance. To evaluate the selection method the Gaussian Mixture Model (GMM) [5] is used as baseline.

The rest of the paper is organized as follows. Section 2 explains the sensitivity of the Gaussian components. Section 3 describes the feature vector selection algorithm. Section 4 shows the results of the experiments. Finally section 5 presents the conclusions and future work.

1.1 Sensitivity of the Gaussian Components

The GMM models the feature vectors of a speech signal, performing a weighted sum of M (number of mixtures of Gaussian probability density functions).

$$p(\hat{x}/\lambda) = \sum_{i=1}^{M} p_i b_i(\hat{x}), \tag{1}$$

We take as input data the MFCC.

$$MFCCMatrix \rightarrow X = \left\{ \begin{array}{cccc} \hat{x}_1 & \hat{x}_2 & \ldots & \hat{x}_T \\ \downarrow & \downarrow & & \downarrow \\ c_{1,1} & c_{1,2} & \ldots & c_{1,T} \\ c_{2,1} & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ c_{D,1} & \ldots & \ldots & c_{D,T} \end{array} \right\} \tag{2}$$

where $\hat{x}_t$ is a feature vector that represents one observation over the signal, t is the index of the speech frame and D is the amount of coefficients. The matrix X is a sequence of random variables indexed by a discrete variable, time ($t = 1, \cdots, T$). Each of the random variables of the process has its own probability distribution function and we assume that they are independent. This is called MFCC matrix and is extracted from a speech expression, which characterizes the speaker.

where: $b_i(\hat{x}_t) \rightarrow$ with $i = 1, \cdots, M$ are the Gaussian density components. Each component is a Gaussian function of the form:

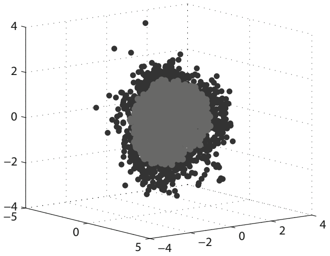


Fig. 1. First three MFCC

$$b_i(\hat{x}) = \frac{1}{(2\Pi)^{\frac{D}{2}}|\Sigma_i^{\frac{1}{2}}|} exp\{-\frac{1}{2}(\hat{x}-\hat{\mu}_i)\acute{}\Sigma_i^{-1}(\hat{x}-\hat{\mu}_i)\} \qquad (3)$$

where: $\mu \rightarrow$ mean matrix, $\Sigma \rightarrow$ covariance matrix.

and: $p_i \rightarrow$ weights of the mixtures $i = 1, 2, \cdots, M$ and satisfies that $\sum_{i=1}^{M} = 1$.

If we represent the first three MFCC (c1, c2, c3) to view their behavior, we would observe a very dense cloud of points toward the center and with some scattered at the edges, the same behavior will follow any group of coefficients that are chosen, for example:

If we generalize the representation of Fig. 1 to D MFCC, we could assume that the D representation would have a very dense cloud of points toward its $D - dimensional$ center.

What would happen if we classify a two dimensional data (c1, c2) or three dimensional data (c1, c2, c3) using 16 Gaussian mixtures?

For the graphical representation of 16 GMM of three MFCC we can conclude:

From this intuitive idea that the individual components of a multi-modal density (GMM) is capable of modeling the underlying acoustic classes in the speech for each speaker, and that speaker's acoustic space can be approximated by a set of acoustic classes (mixtures), we can observe that acoustic classes are more overlapping in the region where the features are more compact. This overlapping reduces the discriminative power of these acoustic classes.

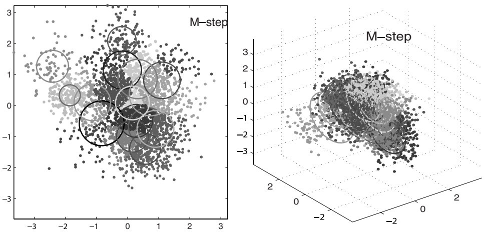


Fig. 2. Classification using 16 Gaussian mixtures of two and three dimensional data

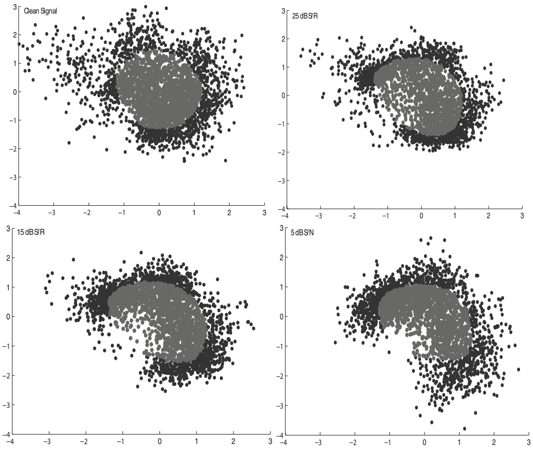


Fig. 3. Representation of the MFCCs distorted by additive white noise: Clean signal, 25db S/N, 15db S/N and 5db S/N

Gaussian components that define the cluster in the dense center of the features are much more overlapped between them, that the Gaussian components that define the features in the border. This makes the probability of Gaussian components given by the features in the dense center more prone to perturbations by the displacements of the features, in presence of noise, as is observed in Fig. 3.

The Fig. 3 shows what happens with the two dimensional coefficients when it is distorted by additive white noise, where it can be clearly seen how the points at the center of the cloud are affected on a larger scale. The intuitive idea that the individual components with less overlapping are capable of modeling the acoustic classes with more robustness in front of the displacements of the features in a noisy speaker verification process motivated us to find an algorithm of feature vector selection capable of only choosing those feature vectors that do not belong to the dense center.

2 Feature Selection Algorithms

From the above we developed an algorithm to select the feature vectors that are outside the dense center to use only these features in the verification process.

We use as input features the MFCC matrix $X_{D,T}$, assuming it describes the speech, then we take each as a point of acoustic space as shown in Fig. 1.

The algorithm can be summarized in four steps:

1. Construct a neighborhood graph - Compute its k neighbours more distant on the acoustic space based on Euclidean distances $d(i, j)$ between pairs of points i, j.

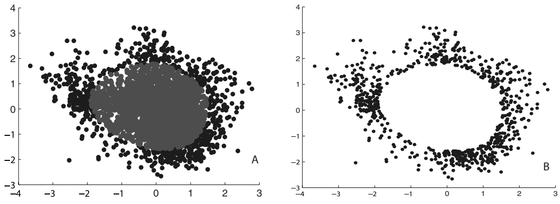


Fig. 4. Results of the feature vector selection (B)

2. Assign weights $W_{i,j}$ to the edges of the graph, in our case, in the $i-th$ row, will have the value 1 those points that belong to the neighboring of the point i, and will have the value 0 those points that are outside these neighboring.
3. Building an array L with length equal to amount of points (T) and in each index is stored the sum of the connections for this node, $y_j = \sum_i W_{i,j}$.
4. Select from the MFCC matrix only the features vectors that correspond to the indices of the previous array (L) which are different from zero. $\overline{X}_{D,V} = X_{D,T}$, where $V << T$.

The Fig. 4A shows the 3000 features vectors of the MFCC matrix. The clusters were defined with 16 neighbors. The Fig. 4B shows selected 826 << 3000 features vectors, all located at the edges, the features vectors located at the dense center were eliminated.

3 Experiments and Results

Ahumada [6] is a speech database of 103 Spanish male speakers, designed and acquired under controlled conditions for speaker characterization and identification. Each speaker in the database expresses six types of utterances in seven microphone sessions and three telephone sessions, with a time interval between them.

The experiment consisted in the evaluation of the performance of feature vector selection in speaker recognition, in front of noisy environment and channel mismatch using spontaneous phrases of 100 speakers in two telephones sessions of Ahumada. The white noise, hf-channel noise and pink noise obtained from Noisex database are artificially added to the samples at SNR ranking from 25 dB, 15 dB and 5 dB.

In order to evaluate the effectiveness of the proposed feature vector selection in speaker recognition, two recognition experiments were implemented for each type of noise and each SNR ranking using 12-dimensional MFCC + delta features vector with Cepstral Mean and Variance normalization applied.

1. Baseline experiment: train with 1 min of spontaneous sentences and test with the 1 min segments, with the SNR ranking applied.
2. Feature vector Selection experiment: the same baseline experiment but in the test phase feature vectors were selected using the proposed method.

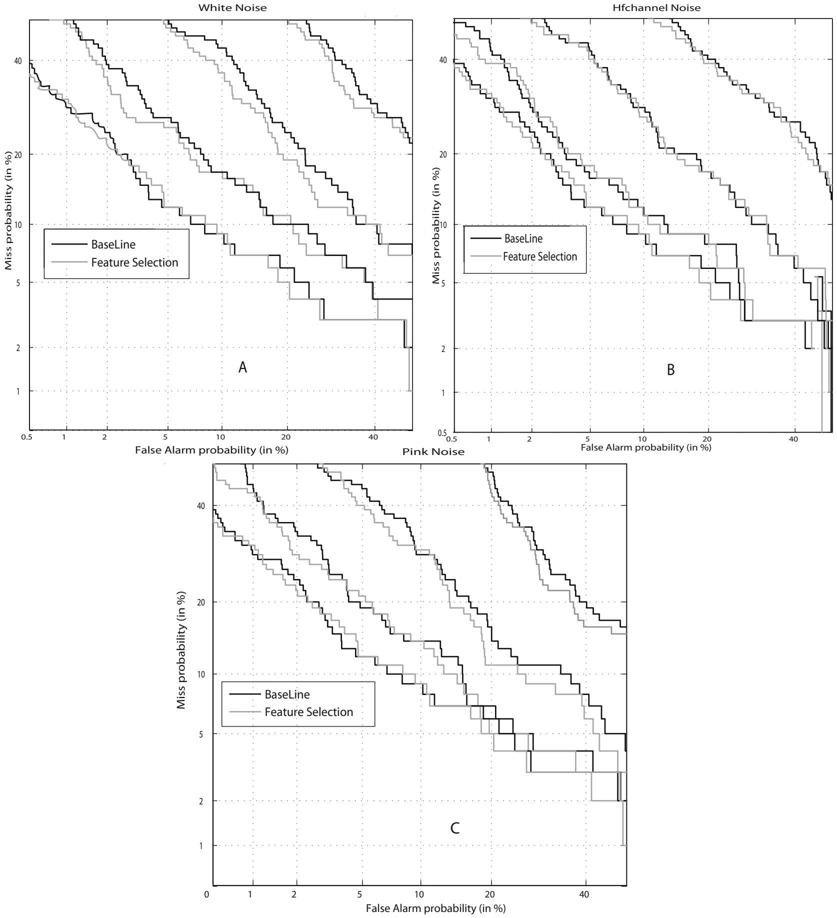


Fig. 5. Speaker recognition DET plots. Black: Baseline. Gray: Feature vector Selection. Clean Signal, 25 dB, 15 dB and 5 dB in the same order that increase the ERR. A) white noise, B) hf-channel noise and C) pink noise.

The performance of both experiments was evaluated using a 16 mixtures GMM classifier [5]. The results of the two experiments are reflected in detection error tradeoff (DET) plot [7], in Fig. 5:

The EER result of the experiments is shown in Table 1.

In the case of white noise (Fig. 5-A), for the first and second DET plot the result are alike, but in the third and fourth DET plot with 15 dB and 5 dB of S/N respectively is appreciable a better behavior using the feature vector selection. In the case of hf-channel noise (Fig. 5-B) the curves are similar in the two experiments for all levels of noise. In the case of pink noise (Fig. 5-C) the results are analogous to white noise, though in the second DET plot we start to view an improvement in the result using the feature vector selection, is valid to note that the white and pink noises are relatives.

Table 1. EER of the experiments

White Noise				
	Signal clean	25 dB SNR	15 dB SNR	5 dB SNR
Baseline	6	15	23	38
Feature Selection	8	15	19	32
Hf-channel Noise				
Baseline	6	13	18	34
Feature Selection	8	10	16	34
Pink Noise				
Baseline	6	15	19	33
Feature Selection	8	13	17	30

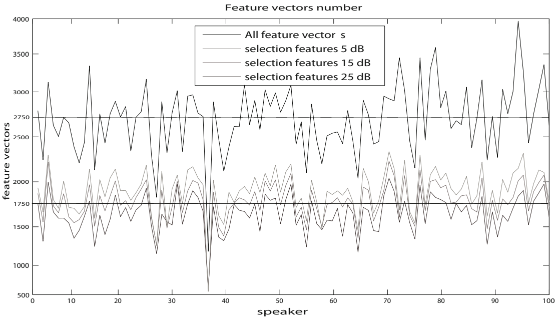


Fig. 6. Number of feature vectors for each speaker and their reduction after making the selection

It empirically shows that the features selected outside dense center are more noise robust for speaker recognition than all features vectors of the MFCC matrix, furthermore this selection allows a smaller amount of feature vectors for recognition. The Fig. 6 shows the difference in the amount of features.

Approximately, 1000 features vectors are eliminated by the selection in each speaker, which represents 20 seconds of each signal; this selection reduces the time calculation of the verification algorithms.

4 Conclusions and Future Work

The experiments results reflect a superior performance of selected MFCC respect to use all the MFCC in speaker recognition using speech samples from telephone sessions of Ahumada Spanish database.

- Results show that speaker recognition in noiseless conditions has the same behavior using either all MFCC or selected MFCC, but with increased noise selected feature show more robustness.
- Tests under noisy conditions (experiments A and C, 15 dB and 25 dB) reflect a better behavior of the selected feature respect to use all MFCC in front of

worst mismatch conditions, (channel and session variability) whereas in the experiment B have a similar behavior.

- In all experiments using the selected feature vectors the computation time of verification algorithms is reduced because of the elimination of 20 secs. from the complete signal.
- Experiments (Table 1) show an EER reduction due to utilization of selected feature vectors instead all MFCC. This reduction is 6 percent in high noisy conditions.

Future work will be in the direction of evaluate the influence of selected feature vectors in other noisy environments.

References

1. Ming, J., Hazen Timothy, J., Glass James, R., Reynolds Douglas, A.: Robust Speaker Recognition in Noisy Conditions. IEEE Trans. on ASLP 15(5) (July 2007)
2. Reynolds, D.A.: Channel robust speaker verication via feature mapping. Proc. of ICASSP, pp. II-53-6 (2003)
3. Teunen, R., Shahshahani, B., Heck, L.: A model-based transformational approach to robust speaker recognition. In: Proc. of ICSLP (2000)
4. Fauve, B.G.B., Matrouf, D., Scheffer, N., Bonastre, J.-F., Mason, J.S.D.: State-of-the-Art Performance in Text-Independent Speaker Verification Through Open-Source Software. IEEE Trans. on ASLP 15(7), 1960–1968 (2007)
5. Douglas, A., Richard, R.y., Rose, C.: Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Trans. on SAP 3(1) (January 1995)
6. Ortega-Garcia, J., Gonzalez-Rodriguez, J., Marrero-Aguiar, V.: AHUMADA A large speech corpus in Spanish for speaker characterization and identification. Speech communication (31), 255–264 (2000)
7. Martin, A., et al.: The DET curve assessment of detection task performance. Proc. of EuroSpeech 4, 1895–1898 (1997)

Implementation of Three Text to Speech Systems for Kurdish Language

Anvar Bahrampour[1], Wafa Barkhoda[2], and Bahram Zahir Azami[2]

[1] Department of Information Technology, Islamic Azad University,
Sanandaj Branch Sanandaj, Iran
bahrampour58@gmail.com
[2] Department of Computer, University of Kurdistan
Sanandaj, Iran
{w.barkhoda,zahir}@ieee.org

Abstract. Nowadays, concatenative method is used in most modern TTS systems to produce artificial speech. The most important challenge in this method is choosing appropriate unit for creating database. This unit must warranty smoothness and high quality speech, and also, creating database for it must reasonable and inexpensive. For example, syllable, phoneme, allophone, and, diphone are appropriate units for all-purpose systems. In this paper, we implemented three synthesis systems for Kurdish language based on syllable, allophone, and diphone and compare their quality using subjective testing.

Keywords: Speech Synthesis; Concatenative Method; Kurdish TTS System; Allophone; Syllable, and Diphone.

1 Introduction

High quality speech synthesis from the electronic form of text has been a focus of research activities during the last two decades, and it has led to an increasing horizon of applications. To mention a few, commercial telephone response systems, natural language computer interface, reading machines for blinds and other aids for the handicapped, language learning systems, multimedia applications, talking books and toys are among the many examples [1].

Most of the existing commercial speech synthesis systems can be classified as either formant synthesizers [2,3] or concatenation synthesizers [4,5]. Formant synthesizers, which are usually controlled by rules, have the advantage of having small footprints at the expense of the quality and naturalness of the synthesized speech [6]. On the other hand, concatenative speech synthesis, using large speech databases, has become popular due to its ability to produce high quality natural speech output [7]. The large footprints of these systems do not present a practical problem for applications where the synthesis engine runs on a server with enough computational power and sufficient storage [7].

Concatenative speech synthesis systems have grown in popularity in recent years. As memory costs have dropped, it has become possible to increase the size of the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 321–328, 2009.

acoustic inventory that can be used in such a system. The first successful concatenative systems were diphone based [8], with only one diphone unit representing each combination of consecutive phones. An important issue for these systems was how to select, offline, the single best unit of each diphone for inclusion in the acoustic inventory [9,10]. More recently there has been interest in automation of the process of creating databases and in allowing multiple instances of particular phones or groups of phones in the database, with the selection decided at run time. A new, but related problem has emerged: that of dynamically choosing the most adequate unit for any particular synthesized utterance [11].

The development and application of text to speech synthesis technology for various languages are growing rapidly [12,13]. Designing a synthesizer for a language is largely dependent on the structure of that language. In addition, there can be variations (dialects) particular to geographic regions. Designing a synthesizer requires significant investigation into the language structure or linguistics of a given region.

In most languages, widespread researches are done on Text-to-Speech systems and also, in some of these languages commercial versions of system are offered. CHATR [14, 15] and AT&T NEXT GEN [16] are two examples offered in English language. Also, in other languages such as French [17,18], Arabic [1,4,7,19,20], Norwegian [21], Korean [22], Greek [23], Persian [24-27], etc, much effort has been done in this field.

The area of Kurdish Text-to-Speech (TTS) is still in its infancy, and compared to other languages, there has been little research carried on in this. To the best of our knowledge, nobody has performed any serious academic research on various branches of Kurdish language processing yet (recognition, synthesis, etc.) [28, 29].

Kurdish is one of the Iranian languages, which are a sub category of the Indian-European family [30]. Kurdish has 24 consonants, 4 semi vowels and 6 vowels. Also /ح/, /ع/, and /غ/ entered Kurdish from Arabic. Also, this language has two scripts: the first one is a modified Arabic alphabet and the second one is a modified Latin alphabet [31]. For example "trifa" which means "moon light" in Kurdish, is written as /تریفه/ in the Arabic script and as "tirîfe" in the Latin. Whereas both scripts are in use, both of them suffer some problems (e.g., in Arabic script the phoneme /i/ is not written; also both /w/ and /u/ are written with the same Arabic written sign /و/ [32,33], and Latin script does not have the Arabic phoneme /ئـ/, and it does not have any standard written sign for foreign phonemes [31]).

In concatenative systems, one of the most important challenges is to select an appropriate unit for concatenation. Each unit has its own advantages and disadvantages, and appropriate for a specific system. In this paper we develop three various concatenative TTS systems for Kurdish language based on Syllable, Allophone, and Diphones, and compare these systems in intelligibility, naturalness, and overall quality.

The rest of the paper is organized as follows: Section 2 introduces the allophone based TTS system. Section 3 and 4 presents syllable and diphone based systems respectively, and finally, comparison between these systems and quality test results are presented in Section 5.

2 Allophone Based TTS System

In this part, a Text-To-Speech system for Kurdish language, which is constructed based on concatenation method of speech synthesis and use allophones (several pronunciation of a phoneme [33]) as basic unit will be introduced[28,29]. According to the input text, proper allophones from database have been chosen and concatenated to obtain the primary output.

Differences between allophones in Kurdish language are normally very clear; therefore, we preferred to explicitly use allophone units for the concatenative method. Some of allophones obey obvious rules; for example if a word end with a voiced phoneme, the phoneme would lose the voicing feature and is called devoiced [34]. However, in most cases there is not a clear and constant rule for all of them. As a result, for extracting allophones we used a neural network. Because their learning power, neural networks can learn from a database and can recognize allophones properly [35].

Fig. 1 shows the architecture of the proposed system. It is composed of three major components: a pre-processing module, a neural network module and an allophone-to-sound module. After converting the input raw text to the standard text, a sliding window of width of four is used as the network input. The network detects second phoneme's allophone, and the allophone waveform is concatenated to the preceding waveform.

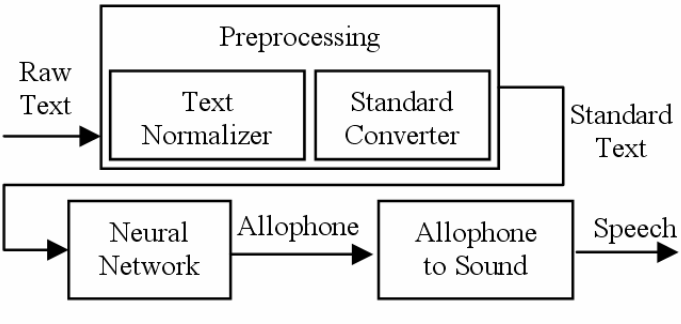


Fig. 1. Architecture of the proposed Kurdish TTS system

The pre-processing module includes a text normalizer and a standard converter. The text normalizer is an application that converts the input text (in Arabic or Latin script) to our standard script; in this conversion we encountered some problems [30,32,34,36].

Finally, in standard script, 41 standard symbols were spotted. Notice that this is more than the number of Kurdish phonemes, because we also include three standard symbols for space, comma and dot. Table 1 shows all the standard letters that are used by the proposed system. Table 2 shows the same sentence in various scripts. Also, the standard converter performs standard text normalization tasks such as converting digits into their word equivalents, spelling out some known abbreviations, etc.

In the next stage, allophones are extracted from the standard text. This task is done using a neural network. Kurdish phonemes have about approximately 200 allophones, but some of them are very similar, and non-expert people cannot detect the differences [34]. As a result, it is not necessary for our TTS system to include all of them (for simplicity, only 66 important and clear instances have been included; see Table 3). Also the allophones are not divided equally between all phonemes (e.g., /p/ is presented by five allophones but /r/ has only one allophone [34]). However, the neural network implementation is very flexible as it is very simple to change the number of allophones or phonemes.

Major Kurdish allophones (more than 80%) are dependent only on the following phonemes. Others (about 20%) are dependent on one preceding and two succeeding phonemes [34]. Hence, we employed four sets of neurons in the input layer, each having 41 neurons for detection of the 41 mentioned standard symbols. A sliding window of width four provides input phonemes for the network input layer. Each set of input layer is responsible for one of the phonemes in the window. The aim is to recognize the relevant allophone to the second phoneme of the window. The output layer has 66 neurons (corresponding to the 66 Kurdish allophones used here) for the recognition of the corresponding allophones and the middle layer is responsible for detecting language rules and it has 60 neurons (These values are obtained empirically); (See Fig. 2). The neural network accuracy rate is equal to 98%. In Table 4, neural network output and desired output are compared.

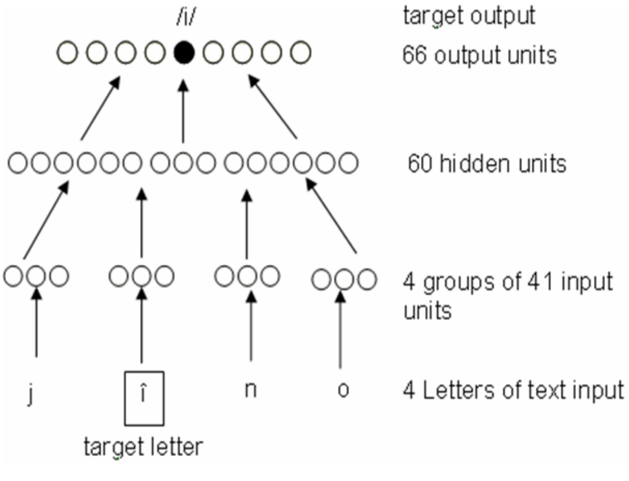


Fig. 2. The neural network structure

After allophone recognition, corresponding waveform of allophones should be concatenated. For each allophone we selected a suitable word and recorded it in a noiseless environment. Separation of allophones in waveforms was done manually by using of Wavesurfer software. The results of this system and comparison between it and other systems are presented in Section 5.

3 Syllable Based TTS System

Syllable is another unit which is used for developing a text-to-speech system. Various languages have different patterns for syllable. In most of these languages, there are many patterns for syllable and therefore, the number of syllables is large; so usually syllable is not used in all-purpose TTS systems. For example, there are more than 15000 syllables in English [6]. Creating a database for this number of units is a very difficult and time-consuming task.

In some languages, the number of syllable patterns is limited, so the number of syllables is small, and creating a database for them is reasonable; therefore this unit can be used in all purpose TTS systems. For example, Indian language has CV, CCV, VC, and CVC syllable patterns, and the total number of syllables in this language is 10000. In [37], some syllable-like units are used; the number of this unit is 1242.

Syllable is used in some Persian TTS systems, too [26]. This language has only CV, CVC, and CVCC patterns for its syllables and so, its syllable number is limited to 4000 [26].

Kurdish has three main groups of syllables that are Asayi, Lekdraw, and Natewaw [36]. Asayi is most the important group and it includes most of the Kurdish syllables. In Lekdraw group, two consonant phonemes occur at the onset of syllable. For example, in /pshu/ two phonemes /p/ and /sh/ make a cluster and the syllable pattern is CCV. Finally, Natewaw group occurs seldom, too. Each group is divided into three groups, Suk, Pir, and Giran [36]. Table 5 shows these groups with corresponding patterns and appropriate examples.

According to Table 5, Kurdish has 9 syllable patterns; but two groups Lekdraw and Natewaw are used seldom and in practice, three patterns, CV, CVC, and CVCC are the most used patterns in Kurdish language. According to this fact, we can consider only Asayi group in implementations, and so the number of database syllables are less than 4000. In our system, we consider these syllables and extend our TTS system using them.

4 Diphone Based TTS System

Nowadays diphone is the most popular unit in synthesis systems. Diphones include a transition part from the first unit to the next unit, and so, have a more desirable quality rather than other units. Also, in some modern systems, a combination of this unit and other methods such as unit selection are used.

Kurdish has 37 phonemes, so in worst case, it has 37×36=1332 diphones. However, all of these combinations are not valid. For example, in Kurdish two phonemes /ع/ and /غ/ or /خ/ and /ح/ do not succeed each other immediately. Also, vowels do not form a cluster. So, the number of serviceable diphones in Kurdish is less than 1300.

After choosing the appropriate unit, we should choose the suitable instance for each unit. For this reason, we chose a proper word for each diphone and then extract its corresponded signal using COOL EDIT. Quality testing results are discussed in Section 5.

5 Quality Testing Results

In this paper, we have developed three synthesis systems based on allophone, syllable, and diphone. In order to assess the quality of the implemented systems and to have a comparison of them, a subjective test was conducted. A set of seven sentences was used as the test material. The test sets were played to 17 volunteer native listeners (5 female, and 12 male). The listeners were asked to rate system's intelligibility, naturalness and overall voice quality on a scale of 1 (bad) to 5 (good). The obtained test results are shown in Table 6.

The allophone based system has the worst quality and in practice, we cannot use it in all-purpose system. In this system we use only 66 most important allophones and so, we can improve its quality using more units in the database.

The syllable based system has intermediate overall quality and high intelligibility. In fact, syllable is a large unit, therefore, its prosody is constant and naturalness is intermediate. On the other hand, because of large size of this unit, this system intelligibility is high.

The diphone based TTS system has best quality between these three systems. Intelligibility and naturalness is high and overall quality is acceptable. Diphones include transition part between a specific phoneme and its next phoneme, so using this unit, we have smooth and pleasant output signal. Hence, diphone is most appropriate unit for developing an all purpose TTS system and in most modern TTS systems use of it as main unit.

Table 1. List of the proposed system standard letters

Arabic	غ	ع	ش	س	ژ	ز	ڕ	ر	د	خ	ح	چ	ج	ت	پ	ب	ا	ئ
Latin	-	-	Ş	s	j	z	rr	r	d	x	-	ç	c	t	p	b	a	-
Standard	X	G	S	s	j	z	R	R	d	x	H	C	c	t	p	b	a	A

Arabic	ی	ی	ه	-	ێ	ه	وو	ۆ	و	و	ن	م	ڵ	ل	گ	ک	ق	ڤ	ف
Latin	y	î	e	i	ê	h	û	o	U	w	n	m	ll	l	g	k	q	v	f
Standard	y	I	e	i	Y	h	U	o	U	w	n	m	L	l	g	k	q	v	f

Table 2. The same sentence in various scripts

Arabic Format	دلوپ دلوپ باران گول ئه نووسيته وه و نمه نمه يش چاوانم تو
Latin Format	Dillop dillop baran gull enûsêtewe û nime nimeyş çawanim to
Standard Format	diLop diLop baran guL AenUsYtewe U nime nimeyS Cawanim to

Table 3. List of phonemes and their corresponding allophones as used in the proposed system

Phoneme	P	b	t	d	K	g	Q	F	s	S	z	J	G	A
Allophones	PpO*&	bEB	t@T	d!D	k?K	G%g	Qq	FVf	s	$S	zZ>	Jj	^	A

Phoneme	C	h	H	m	X	X	n	v	y	l	r	L	R	Y
Allophones	Cc	h	H	mWM	X	X	nN	v	y	l	r	L	R	Y

Phoneme	E	a	N	U	u	o	w	I	i	C	Ü	.	,
Allophones	E	a	#	U	u	o	w	I	i	~	_	.	,

Table 4. A comparison between neural network output and desired output

NN Output	DiLo&_DiLoP_baraN_GuL_AenUsY@_U,_nime_nimeyS_~awaniM_to
Desired Output	DiLo&_DiLo&_baraN_GuL_AenUsY@_U,_nime_nimeyS_~awaniM_to

Table 5. Kurdish syllable patterns

		Suk	Pir	Giran
Asayi	Syllable Pattern	CV	CVC	CVCC
	Example	De, To	Waz, Lix	Kurt, Berd
Lekdraw	Syllable Pattern	CCV	CCVC	CCVCC
	Example	Bro, Chya	Bjar, Bzut	Xuast, Bnesht
Natewaw	Syllable Pattern	V	VC	VCC
	Example	-i	-an	-and

Table 6. Subjective testing results for various systems

	Intelligibility	Naturalness	Overall Quality
Allophone Based TTS System	2.71	2.31	2.45
Syllable Based TTS System	3.35	2.85	3.02
Diphone Based TTS System	3.9	3.37	3.51

References

1. Al-Muhtaseb, H., Elshafei, M., Al-Ghamdi, M.: Techniques for High Quality Arabic Speech Synthesis. In: Information sciences. Elsevier Press, Amsterdam (2002)
2. Styger, T., Keller, E.: Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts. In: Keller, E. (ed.) State of the Art, and Future Challenges Formant synthesis, pp. 109–128. John Wiley, Chichester (1994)
3. Klatt, D.H.: Software for a Cascade/Parallel Formant Synthesizer. Journal of the Acoustical Society of America 67, 971–995 (1980)
4. Hamza, W.: Arabic Speech Synthesis Using Large Speech Database. PhD. thesis, Cairo University, Electronics and Communications Engineering Department (2000)
5. Donovan, R.E.: Trainable Speech Synthesis. PhD. thesis, Cambridge University, Engineering Department (1996)
6. Lemmetty, S.: Review of Speech Synthesis Technology. M.Sc Thesis, Helsinki University of Technology, Department of Electrical and Communications Engineering (1999)
7. Youssef, A., et al.: An Arabic TTS System Based on the IBM Trainable Speech Synthesizer. In: Le traitement automatique de l'arabe, JEP–TALN 2004, Fès (2004)
8. Olive, J.P.: Rule synthesis of speech from diadic units. In: ICASSP, pp. 568–570 (1977)
9. Syrdal, A.: Development of a female voice for a concatenative text-to-speech synthesis system. Current Topics in Acoust. Res. 1, 169–181 (1994)
10. Olive, J., van Santen, J., Moebius, B., Shih, C.: Multilingual Text-to-Speech Synthesis: The Bell Labs Approach, pp. 191–228. Kluwer Academic Publishers, Norwell (1998)
11. Beutnagel, M., Conkie, A., Syrdal, A.K.: Diphone Synthesis using Unit Selection. In: The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis, ISCA (1998)
12. Sproat, R., Hu, J., Chen, H.: Emu: An e-mail preprocessor for text-to-speech. In: Proc. IEEE Workshop on Multimedia Signal Proc., pp. 239–244 (1998)
13. Wu, C.H., Chen, J.H.: Speech Activated Telephony Email Reader (SATER) Based on Speaker Verification and Text-to- Speech Conversion. IEEE Trans. Consumer Electronics 43(3), 707–716 (1997)
14. Black, A.: CHATR, Version 0.8, a generic speech synthesis, System documentation. ATR-Interpreting Telecommunications Laboratories, Kyoto, Japan (1996)

15. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: ICASSP, vol. 1, pp. 373–376 (1996)
16. Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., Syrdal, A.: The AT&T NEXT-GEN TTS System. In: Joint Meeting of ASA, EAA, and DAGA (1999)
17. Dutoit, T.: High Quality Text-To-Speech Synthesis of the French Language. Ph.D. dissertation, submitted at the Faculté Polytechnique de Mons (1993)
18. Dutoit, T., et al.: The MBROLA project: towards a set of high quality speech synthesizers free of use of non commercial purposes. In: ICSLP 1996, Proceedings, Fourth International Conference, IEEE (1996)
19. Chouireb, F., Guerti, M., Naïl, M., Dimeh, Y.: Development of a Prosodic Database for Standard Arabic. Arabian Journal for Science and Engineering (2007)
20. Ramsay, A., Mansour, H.: Towards including prosody in a text-to-speech system for modern standard Arabic. In: Computer Speech & Language. Elsevier, Amsterdam (2008)
21. Amdal, I., Svendsen, T.: A Speech Synthesis Corpus for Norwegian. In: lrec 2006 (2006)
22. Yoon, K.: A prosodic phrasing model for a Korean text-to-speech synthesis system. In: Computer Speech & Language, Elsevier, Amsterdam (2006)
23. Zervas, P., Potamitis, I., Fakotakis, N., Kokkinakis, G.: A Greek TTS based on Non uniform unit concatenation and the utilization of Festival architecture. In: First Balkan Conference on Informatics, Thessalonica, Greece, pp. 662–668 (2003)
24. Farrohki, A., Ghaemmaghami, S., Sheikhan, M.: Estimation of Prosodic Information for Persian Text-to-Speech System Using a Recurrent Neural Network. In: ISCA, Speech Prosody 2004, International Conference (2004)
25. Namnabat, M., Homayunpoor, M.M.: Letter-to-Sound in Persian Language Using Multy Layer Perceptron Neural Network. Iranian Electrical and Computer Engineering Journal (2006) (in persian)
26. Abutalebi, H.R., Bijankhan, M.: Implementation of a Text-toSpeech System for Farsi Language. In: Sixth International Conference on Spoken Language Processing (2000)
27. Hendessi, F., Ghayoori, A., Gulliver, T.A.: A Speech Synthesizer for Persian Text Using a Neural Network with a Smooth Ergodic HMM. ACM Transactions on Asian Language Information Processing, TALIP (2005)
28. Daneshfar, f., Barkhoda, W., Azami, B.Z.: Implementation of a Text-to-Speech System for Kurdish Language. In: ICDT 2009, Colmar, France (2009)
29. Barkhoda, W., Daneshfar, F., Azami, B.Z.: Design and Implementation of a Kurdish TTS System Based on Allophones Using Neural Network. In: ISCEE 2008, Zanjan, Iran (2008) (in persian)
30. Thackston, W.M.: Sorani Kurdish: A Reference Grammar with Selected Reading. Iranian Studies at Harvard University, Harvard (2006)
31. Sejnowski, J.T., Rosenberg, R.: Parallel Networks that Learn to Pronounce English Text, pp. 145–168. The Johns Hopkins University, Complex Systems Inc. (1987)
32. Rokhzadi, A.: Kurdish Phonetics and Grammar. Tarfarnd press, Tehran (2000)
33. Deller, R.J., et al.: Discrete time processing of speech signals. John Wiley and Sons, Chichester (2000)
34. Kaveh, M.: Kurdish Linguistic and Grammar (Saqizi accent), 1st edn. Ehsan Press, Tehran (2005) (In Persian)
35. Karaali, O., et al.: A High Quality Text-to-Speech System Composed of Multiple Neural Networks. In: Invited paper, IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle (1998)
36. Baban, S.: Phonology and Syllabication in Kurdish Language, 1st edn. Kurdish Academy Press, Arbil (2005) (In Kurdish)
37. Rao, M.N., Thomas, S., Nagarajan, T., Murthy, H.A.: Text-to-Speech Synthesis using syllable-like units. In: National Conference on Communication, India (2005)

Functional Feature Selection by Weighted Projections in Pathological Voice Detection*

Luis Sánchez Giraldo[1,**], Fernando Martínez Tabares[2], and Germán Castellanos Domínguez[2]

[1] University of Florida, Gainesville, FL, 32611, USA
luisitobarcito@ufl.edu

[2] Universidad Nacional de Colombia Km 7 vía al Magdalena, Manizales, Colombia
{fmartinezt,cgcastellanosd}@unal.edu.co

Abstract. In this paper, we introduce an adaptation of a multivariate feature selection method to deal with functional features. In our case, observations are described by a set of functions defined over a common domain (e.g. a time interval). The feature selection method consists on combining variable weighting with a feature extraction projection. Although the employed method was primarily intended for observations described by vectors in $\mathbb{R}^n$, we propose a simple extension that allows us to select a set of functional features, which is well suited for classification. This study is complemented by the incorporation of Functional Principal Component Analysis (FPCA) that project functions into a finite dimensional space were we can perform classification easily. Another remarkable property of FPCA is that it can provide insight about the nature of the functional features. The proposed algorithms are tested on a pathological voice detection task. Two databases are considered: Massachusetts Eye and Ear Infirmary Voice Laboratory voice disorders database and Universidad Politécnica de Madrid voice database. As a result, we obtain a canonical function whose time average is enough to reach similar performances to the ones reported in the literature.

1 Introduction

Pattern recognition from the side of machine learning is more concerned with rather general methods for extracting information from the available data, and thence the task of handcrafting complex features for each individual problem turns to be less crucial. Consequently, large sets of simpler features are employed, however, since these variables are less refined for each problem they require of some processing. Feature selection has been discussed in the past by several authors [1,2] as an important preprocessing stage to improve results during and after training in learning processes that also attempts to overcome the curse of dimensionality. More recent studies on this subject are found in [3,4]. A common issue in machine learning approach takes place when the number

* This is part of the project 20201004208, funded by Universidad Nacional de Colombia.

** The author was a student at Universidad Nacional de Colombia.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 329–336, 2009.

relevant features is considerably smaller and the computation time, that grows exponentially with the number of features, becomes prohibitively large for real time applications [5]. Yet there is a more fundamental issue related to interpretation, when there is a large amount of incoming data. Reducing the size of data either by encoding or removing irrelevant information, becomes necessary if one wants to achieve good performance in the system as well as insightful results. In our work, we attempt to combine feature selection and extraction with a twofold purpose: interpretation and generalization.

Functional data analysis can be regarded as an extension of the existing multivariate methods to more involved representations. The typical framework in multivariate statistics deals with descriptions of the objects as vectors in $\mathbb{R}^n$. In the case of functional data, we have a set of functions with the same domain that are extracted from each observation. Even if these functions are in a discrete domain, the dimensionality of data poses a challenging problem. Ramsay and Silverman [6] give a thorough presentation of the topic. Methods such as PCA on functional data can be found in [7], and nonparametric extensions are discussed in [8]. Despite there is clear interest on developing methods or adaptations for this functional representations, none of these works address the problem of selecting the functions that might provide the relevant information for the analysis; what is more, most of the analysis focuses on objects described by a single function. Even though the authors may argue the extension of these methods to several functions is straightforward, there must be some concern on the choice of the set of functions that ought be analyzed. Having irrelevant functions describing the problem may hinder the effect of the relevant ones.

In this work, we use a weighting method presented in [9] for attaining subset selection. The method combines feature selection and feature extraction on the same process. Particularly, we employ Weighted Regularized Discriminant Analysis (WRDA), which allows us to obtain a rather simple generalization to functional data. The optimization process consists on maximizing a trace ratio over a fixed number of discriminant directions. The paper starts with the description of the employed method for the case of regular features (vectors in $\mathbb{R}^n$). Then, we describe the adaptation process for functional data. In order to observe the effectiveness of the proposed adaptation for the weighting algorithm, tests on Pathological voice databases were carried out. Two databases are considered: Massachusetts Eye and Ear Infirmary Voice Laboratory voice disorders database distributed by Kay Elemetrics Corp. (KLM) and the Universidad Politécnica de Madrid voice disorders database (UPM).

2 Methods

Variable selection consist basically on selecting a subset of features from a larger set. This type of search is driven by some evaluation function that have been defined as the relevance of a given set [2]. When this process implies exhaustive search (binary selection), the relevance measure must take into account the dimensionality. Feature weighting relax this constraint allowing the calculation of derivatives of the target function or the use mathematical programming tools to optimize these weights[10]. One important point is that the weights of the irrelevant variables should be as close as possible to zero, and the weights of the relevant features should be bounded. On the other hand. Feature extraction aims at encoding data efficiently for the problem at hand. In the case of linear projections for feature extraction, we can capitalize on this property to keep a fixed dimension (projected space) and assess the relevancy of the projected set. Thence, we can combine feature extraction methods with weighted data to maintain a fixed set size and accommodate weights in such a manner a relevance criterion is somehow maximized. Surprisingly, this consideration plays crucial role in guaranteeing that low weights will vanish.

2.1 Regularized Discriminant Analysis

RDA was proposed by [11] for small sample, high dimensional data sets to overcome the degradation of the discriminant rule. The aim of the linear variant of this technique is to find a projection of the space where scatter between classes is maximized maintaining the within scatter as minimal as posible. This is achieved by maximizing the ratio between the projected between class and within class matrices $J = \frac{|\mathbf{U}^T \boldsymbol{\Sigma}_B \mathbf{U}|}{|\mathbf{U}^T \boldsymbol{\Sigma}_W \mathbf{U}|}$, where $\mathbf{U}$ is the projection matrix whose dimension is given by the number of classes (k) to be linearly separated, $\boldsymbol{\Sigma}_B$ is the between class matrix and can be associated to the dispersion of the mean values of each class, and $\boldsymbol{\Sigma}_W$ is the within class matrix and can be linked to the average class covariance matrix. The problem is defined as the constrained maximization of $|\mathbf{U}^T \boldsymbol{\Sigma}_B \mathbf{U}|$, that is, $\max_{\mathbf{U}} |\mathbf{U}^T \boldsymbol{\Sigma}_B \mathbf{U}|, \text{ subject to } |\mathbf{U}^T \boldsymbol{\Sigma}_W \mathbf{U}| = 1$. Conditional extremes can be obtained from Lagrange multipliers; the solutions are the $k-1$ leading generalized eigenvectors of $\boldsymbol{\Sigma}_B$ and $\boldsymbol{\Sigma}_W$ that are the leading eigenvectors of $\boldsymbol{\Sigma}_W^{-1} \boldsymbol{\Sigma}_B$. The need of regularization arises from small samples were $\boldsymbol{\Sigma}_W$ can not be directly inverted. The solution is rewritten as:

$$(\boldsymbol{\Sigma}_W + \delta \mathbf{I})^{-1} \boldsymbol{\Sigma}_B \mathbf{U} = \mathbf{U} \boldsymbol{\Lambda}. \tag{1}$$

After weighting data, that is $\mathbf{XD}$ where $\mathbf{D}$ is diagonal matrix with ideally zero entries corresponding to the irrelevant features, J becomes $J_{\mathbf{D}} = \frac{|\mathbf{U}^T \mathbf{D} \boldsymbol{\Sigma}_B \mathbf{D} \mathbf{U}|}{|\mathbf{U}^T \mathbf{D} \boldsymbol{\Sigma}_W \mathbf{D} \mathbf{U}|}$.

2.2 Variable Weighting and Relevance Criterion

Data will be projected onto a fixed dimension subspace. For WRDA the fixed dimension should range between 1 to $k-1$; being k the number of classes. This consideration depends on the distribution of the classes within the subspace. The relevance of a given weighted projection of a fixed dimension is evaluated by a separability measure. The search function will fall in some local maximum of the target function. The parameter to be optimized is the weight matrix $\mathbf{D}$, and selected criterion is the ratio traces of the aforementioned within and between matrices; this criterion is called J_4 in [12]. For weighted-projected data this measure is given by: $J_4(\mathbf{D}, \mathbf{U}) = \frac{\text{trace}(\mathbf{U}^T \mathbf{D} \boldsymbol{\Sigma}_B \mathbf{D} \mathbf{U})}{\text{trace}(\mathbf{U}^T \mathbf{D} \boldsymbol{\Sigma}_W \mathbf{D} \mathbf{U})}$ The size of $\mathbf{U}$ is $(c \times p)$ and p denotes the

Algorithm 1. WRDA

```
1: Set dimension $p = k - 1$, being $k$ the number of classes
2: Normalize each feature vector to have zero mean and $\| \cdot \|_2 = 1$
3: Start with some initial set of orthonormal vectors $\mathbf{U}^{(0)}$
4: Compute $\mathbf{d}^{(r)}$ from solution given in section 2.2, and reweigh data.
5: Compute the $\mathbf{U}^{(r)}$ from solution given in section 2.1.
6: Compare $\mathbf{U}^{(r)}$ and $\mathbf{U}^{(r-1)}$ for some $\varepsilon$ and return to step 3 if necessary.
```

We make use of the sum of absolute values of the diagonal elements of $\left(\mathbf{U}^{(r)}\right)^T \mathbf{U}^{(r-1)}$, which are compared to the value obtained for $\left(\mathbf{U}^{(r-1)}\right)^T \mathbf{U}^{(r-2)}$

fixed dimension, which is the number of projection vectors $\mathbf{U} = \left(\mathbf{u}_1 \; \mathbf{u}_2 \; \cdots \; \mathbf{u}_f \right)$. In order to apply matrix derivatives conveniently, we may want to rewrite $\mathbf{D}$ in terms of its diagonal entries and represent it as a column vector $\mathbf{d}$. For this purpose, we can use the identity trace $\left(\mathbf{U}^T\mathbf{DKDU}\right) = \mathbf{d}^T \left(\sum_{i=1}^{p} \mathbf{K} \circ \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{d}$ to rewrite the trace ratio can be rewritten in terms of Hadamard products as

$$J_4(\mathbf{d}) = \left\{ \mathbf{d}^T \left(\sum_{i=1}^{p} \boldsymbol{\Sigma}_B \circ \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{d} \right\} / \left\{ \mathbf{d}^T \left(\sum_{i=1}^{p} \boldsymbol{\Sigma}_W \circ \mathbf{u}_i \mathbf{u}_i^T \right) \mathbf{d} \right\} \tag{2}$$

This target function is quite similar to the one obtained for the regularized LDA. Therefore, the solution of $\mathbf{d}$ with constrained L^2 norm is given by the leading eigenvector of

$$\left(\sum_{i=1}^{p} \boldsymbol{\Sigma}_W \circ \mathbf{u}_i \mathbf{u}_i^T + \delta \mathbf{I} \right)^{-1} \left(\sum_{i=1}^{p} \boldsymbol{\Sigma}_B \circ \mathbf{u}_i \mathbf{u}_i^T \right) \tag{3}$$

3 Functional Data Adaptation

Let $\mathcal{X}$ be a set of objects that we want to classify into k different classes. Each observation $x \in \mathcal{X}$ is represented by a c-tuple of functions defined in the same domain, for instance $x = (f_1, \; f_2, \; \cdots, \; f_c)$ and $f_l \in L_2[a,b]$, for $l = 1, \; \cdots, \; c$. If we want to plug in the feature selection algorithm 1, presented in the previous sections, we need to define a way of quantifying the variation in the space of real square integrable functions $L_2[a,b]$. To this end, we will define the following key elements: **the expected function** $E[f_l(t)] = \int_{\mathbb{R}} f dF_l(f;t)$, **the expected squared norm** $E[\|f_l\|^2] = \int_a^b \left(\int_{\mathbb{R}} |f|^2 dF_l(f;t) \right) dt$, **the expected inner product** $E[\langle f_l, g_m \rangle] = \int_a^b \left(\iint_{\mathbb{R}} (fg) dF_{lm}(f,g;t) \right) dt$; where $F_l(f;t)$ is the first-order probability distribution of l-th stochastic process represented by $f_l(t,x)$, and $F_lm(f,g;t)$ is the joint probability distribution of the l-th and m-th stochastic processes $f_l(t,x)$ and $f_m(t,x)$. In general, we just have access to a discrete version $f_l[t]$ of the function f_l; besides, $F_l(f;t)$ and $F_{lm}(f,g;t)$ are unknown. The only available information is provided by the sample $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i = (f_{1i}[t], \; f_{2i}[t], \; \cdots, \; f_{ci}[t])$ for $1 \leq t \leq T$, and $y_i \in \mathcal{Y} = \{1, \; 2, \; \cdots, \; k\}$ is the class label for the observed x_i. Under these conditions we define the discrete empirical estimations of the expected values: $E_{\text{emp}}[f_l[t]] = \frac{1}{n}\sum_{i=1}^{n} f_{li}[t]$, $E_{\text{emp}}[\|f_l\|^2] = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{t=1}^{T}|f_{li}[t]|^2\right)$, and $E_{\text{emp}}[\langle f_l, f_m\rangle] = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{t=1}^{T} f_{li}[t]f_{mi}[t]\right)$. With this elements we can construct analogs for $\boldsymbol{\Sigma}_W$ and $\boldsymbol{\Sigma}_B$. Notice that Algorithm 1, requires of a previous normalization of data. In the functional case this can be achieved by removing to each observation x_i the overall empirical mean of the sample, that is, $\widehat{f}_{li}[t] = f_{li}[t] - E_{\text{emp}}[f_l[t]] \quad \text{for} \quad 1 \le l \le c$, and scaling the values that each function takes $\widetilde{f}_{li}[t] = \frac{\widehat{f}_{li}[t]}{\sqrt{nE_{\text{emp}}[\|\widehat{f}_l[t]\|^2]}} \quad \text{for} \quad 1 \le l \le c$. From now and on, to ease the notation, we will assume that $f_{li}[i]$ is the normalized version of the function, that is $E_{\text{emp}}[f_l[t]] = 0$ and $E_{\text{emp}}[\|f_l\|^2] = 1/n$. For each j class, we define the empirical class-conditional expected values, which are computed as follows:

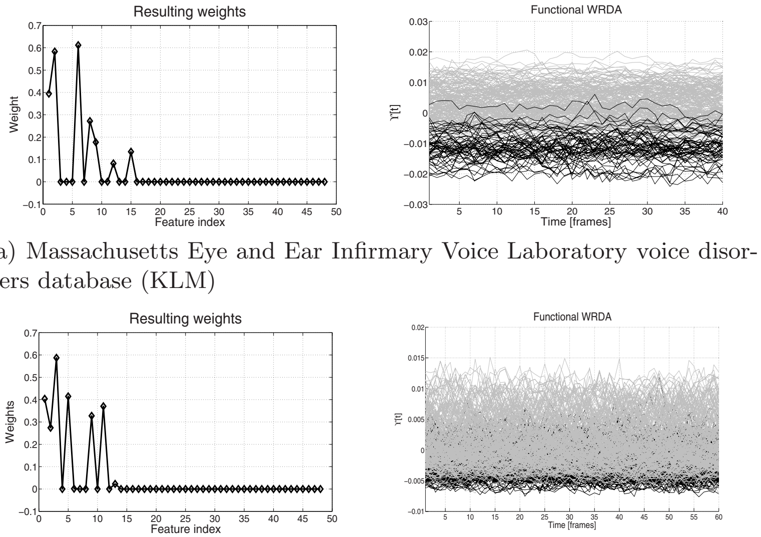
$$E_{\text{emp}}[T\{f_l\}|j] = \frac{1}{n_j}\sum_{x_i|y_i=j} T\{f_{li}\} \qquad E_{\text{emp}}[T\{f_l, f_m\}|j] = \frac{1}{n_j}\sum_{x_i|y_i=j} T\{f_{li}, f_{mi}\} \quad (4)$$

where n_j is the number of observations that belong to j-th class, $T\{\cdot\}$ and $T\{\cdot,\cdot\}$ are functions over f. The j-th within class matrix $\boldsymbol{\Sigma}_{Wj}$ has the following elements $wj_{lm} = E_{\text{emp}}[\langle f_l - E_{\text{emp}}[f_l|j], f_m - E_{\text{emp}}[f_m|j]\rangle|j]$, and the pooled within class matrix $\boldsymbol{\Sigma}_W$ cam be computed as $\boldsymbol{\Sigma}_W = \sum_{j=1}^{k} n_j\boldsymbol{\Sigma}_{Wj}$. The between class matrix $\boldsymbol{\Sigma}_B$ elements are $b_{lm} = \sum_{j=1}^{k} n_j\langle E_{\text{emp}}[f_l|j], E_{\text{emp}}[f_l|j]\rangle$. Once $\boldsymbol{\Sigma}_W$ and $\boldsymbol{\Sigma}_B$ have been obtained, we can proceed with the rest of Algorithm 1, normally.

4 Experiments and Discussion

We refer to [13] for a complete description of KLM and UPM databases. Functional features correspond to windowed estimations of Harmonic Noise Ratio*HNR*, Normalized Noise Energy *NNE*, Glottal Noise Energy *GNE*, Energy, and 12 Mel Freq Cepstral Coefficients along with their first and second order derivatives obtained as in [14] for a total 48 functional features. These time vectors were clipped to a fixed number of windows moving from the central window to the sides, symmetrically. The fixed length of the sampled functions was 40 instances per functional feature in KLM, and 60 in UPM. The preliminary analysis consists on finding a set p of canonical functions resulting from a linear combination of the original set of c functional features using the Functional WRDA algorithm, $\Upsilon_{ji}[t] = \sum_{l=1}^{c}\alpha_{jl}f_{li}[t]$.where α_{jl} is obtained from the entries of the weighting and rotation matrices $\mathbf{D} = \text{diag}(d_1, d_2, \cdots, d_l)$ and $\mathbf{U} = \begin{pmatrix}\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_p\end{pmatrix}$, and i is the index for the i-th observation.

In the two class case, which is our case Pathological vs Normal, the set of canonical functions reduces to a single function, that is, $\mathbf{U} = \mathbf{u}_1$. The regularization parameter δ introduced in equations (1) and (3) was set to 0.137 for KLM and 0.12 for UPM. Some graphical results for both databases are depicted in Figure 1. Right plots form Figures 1(a) and 1(b) present the weighted linear combination of the original functional features (canonical function) for KLM and UPM databases. In both databases most of the zero weights correspond to the first and second derivatives of the original features. The right plots show



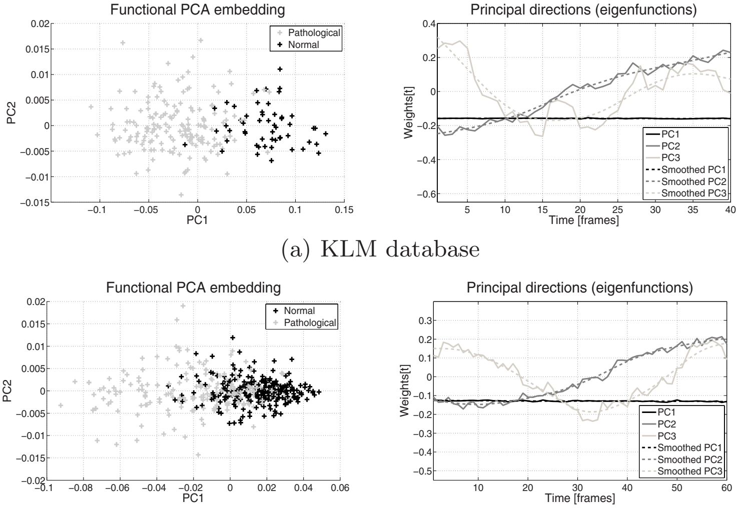
(a) Massachusetts Eye and Ear Infirmary Voice Laboratory voice disorders database (KLM)

(b) Universidad Politécnica de Madrid voice disorders database (UPM)

Fig. 1. Weights and resulting canonical functions for KLM and UPM databases. Gray lines are canonical functions from pathological examples and black lines are from normal observations. The left plots are the resulting weights for each one of the functional features. The first 16 indexes are the short term energy and MFCC features. Notice that indexes from 17 to 48 obtained zero weights; these indexes correspond to first and second order derivatives of the functional features.

the resulting canonical functions from the whole set of examples for the two databases.

It is possible to use these functional features to perform classification with a kernel classifier or a distance based classifier by simply computing the inner product or the Euclidean distance between pairs of observations that now are represented by a canonical function. In here, we use Functional PCA (FPCA)[6] to embed this canonical function into a smaller dimension Euclidean space and then perform discrimination with a pooled covariance matrix classifier whose decision function is linear. This approach is equivalent to Kernel PCA [15] using the inner product $k(x_i, x_j) = \langle \Upsilon_{1i}[t], \Upsilon_{1j}[t] \rangle = \sum_{l=1}^{c} \sum_{r=1}^{c} \alpha_{1l}\alpha_{1r} \langle f_{li}[t], f_{rj}[t] \rangle$. Figures 2(a) and 2(b) display the clustered classes and how the first principal component may suffice for accurate classification of the sample. Moreover, the shape of the first principal functions and how points are distributed in the embedding suggest a particular phenomenon. The first principal function for both databases is approximate constant, so the inner product between the canonical function and the first PC is equivalent to a time average of the canonical function, which in turn is a sum of time averages of the selected original functional features (features with non-zero weights). PCA result seem to coincide with a LDA projection; a particular situation when both methods coincide is for a two class problem where the within class covariance functions are isotropic



(a) KLM database

(b) UPM database

Fig. 2. Functional PCA embedding using the first two principal components and principal functions for KLM and UPM databases. Right plots are the principal functions for both databases. Notice how the functions are very similar, even though, the origin of the databases differ. The overlap of classes is higher for UPM database; this might be due to the larger diversity in the pathologies for this database.

approximately equal with a significant difference between their means. At the same time we carry out FPCA over the whole set of original functions. We employ a linear classifier using a pooled covariance matrix. Table 1, exhibit the Leave-One-Out (LOO) training and test errors for 1 to 3 principal components after applying the proposed functional feature selection process. This values are contrasted with the LOO training and test errors for 1, 10, and 20 principal components obtained from FPCA of the normalized original data. Our method conveys almost the same error estimate when varying the number of components. In the second case, we obtain incremental performance on training, but it should be noted that as dimensionality grows, so does the confidence interval.

Table 1. LOO training and test errors for FPCA after functional WRDA and FPCA for normalized original data. Errors are remarkably stable for the proposed method.

Database		FWRDA and FPCA			FPCA		
		1PC	2PCs	3PCs	1PC	10PCs	20PCs
KLM	train	9.09	9.23	8.15	13.03	7.78	7.0
	test	9.95	10.41	10.41	12.22	8.14	7.24
UPM	train	24.65	25.08	24.65	40.89	25.21	22,79
	test	25.91	26.14	25.91	40.91	27.27	25.00

5 Conclusions

We have presented a functional feature selection criterion based on weighting variables followed by a projection onto a fixed dimension subspace. Results showed how reducing dimensionality benefits the overall performance of the inference system. The canonical function devised from the application of our method was decomposed using FPCA, an interesting result of this analysis is that time averages can provide the necessary information to carry out successful classification. It is also important to highlight that the set of functional features selected for each of the databases is very similar. Although, both databases are voice disorder databases, their origins are quite different. The similarity of the results is also confirmed by observing the principal functions for both databases.

References

1. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: ICML (1994)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. In: AI, vol. 97(1-2) (1997)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. In: JMLR (2003)
4. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. In: JMLR (2004)
5. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. In: JMLR (2005)
6. Ramsay, J., Silverman, B.: Functional Data Analysis, 2nd edn. Springer, Heidelberg (2005)
7. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
8. Ferraty, F., Vieu, P.: Nonparametric Functional Data Analysis. Springer, Heidelberg (2006)
9. Sánchez, L., Martínez, F., Castellanos, G., Salazar, A.: Feature extraction of weighted data for implicit variable selection. In: CAIP. Springer, Heidelberg (2007)
10. Bradley, P.S., Mangasarian, O.L., Street, W.N.: Feature selection via mathematical programming. INFORMS Journal on Computing 10 (1998)
11. Friedman, J.H.: Regularized discriminant analysis. Journal of the American Statistical Association (1989)
12. Webb, A.R.: Statistical Pattern Recognition, 2nd edn. John Wiley & Sons, Chichester (2002)
13. Daza, G., Arias, J., Godino, J., Sáenz, N., Osma, V., Castellanos, G.: Dynamic feature extraction: An application to voice pathology detection. Intelligent Automation and Soft Computing 15(4) (2009)
14. Godino-Llorente, J.I., Gómez-Vilda, P., Blanco-Velasco, M.: Dimensionality reduction of pathological voice quality assesment system based on gaussian mixtures models and short-term cepstarl parameters. IEEE Transactions on Biomedical Engineering 53(10), 1943–1953 (2006)
15. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, December 2002. MIT Press, Cambridge (2002)

Prediction of Sequential Values for Debt Recovery

Tomasz Kajdanowicz and Przemysław Kazienko

Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
{tomasz.kajdanowicz,kazienko}@pwr.wroc.pl

Abstract. The concept of new approach for debt portfolio pattern recognition is presented in the paper. Aggregated prediction of sequential repayment values over time for a set of claims is performed by means of hybrid combination of various machine learning techniques, including clustering of references, model selection and enrichment of input variables with prediction outputs from preceding periods. Experimental studies on real data revealed usefulness of the proposed approach for claim appraisals. The average accuracy was over 93%, much higher than for simplifier methods.

Keywords: financial pattern recognition, prediction, repayment prediction, claim appraisal, competence regions modeling.

1 Introduction

Valuation of the debt portfolio is a prediction task that assesses the possible repayment value from the debt cases. All business models that rely on the cash flow from receivables assume to minimize the aggregated debt value. The fact that possession of creditors for a long time is very ineffective, especially when debtors are eventually not able to repay theirs arrears in short term, implies a need of sophisticated debts valuation. Under some circumstances, it is better for companies to sell the liabilities to a specialized debt collection company in order to obtain at least a part of their nominal value, rather than collect and vindicate debts on their own. In the process of selling a debt portfolio the transaction price is usually estimated based on the possible repayment level to be reached in the long term. In general, it is expected that the method of debt portfolio value appraisal will well match the future.

2 Related Work

There exists a wide variety of studies on prediction and classification in the literature e.g. [2, 12, 16]. Overall, the existing machine learning methods usually provide better classification and prediction accuracy than techniques based only on common statistical techniques such as regression [17, 18]. A better precision of the prediction may be obtained by combination of several existing methods into one hybrid solution [1, 2, 4]. In general, hybridization could be achieved either by application of additional external mechanisms into existing prediction models (low level), e.g. neuro-fuzzy systems [11] or by combination of different methods on the high level, e.g. multiple

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 337–344, 2009.

classifier systems, where separate classifiers are treated more likewise ‘black boxes’ [6, 7]. Hybrid prediction methods have been successfully used in a number of domains such as medicine, engineering and industry. Other application areas of these methods are economy and finance, where hybrid systems provide specialized knowledge in order to support business decisions [15].

The paper is focused on the description of a new hybrid method for debt portfolio appraisal. The correct prediction of target repayment value in debt recovery is of great practical importance, because it reveals the level of possible expected benefit and chances to collect receivables. The crucial concept of this method is the combination of clustering of the training set and application of multiple classifiers based on their competence region[12]. Additionally, a sequence of classifiers is built to obtain predictions over consecutive periods. Apart from the general idea, the proposed hybrid prediction method has been examined on real data. According to the findings achieved, the method appears to return more precise results compared to some common approaches.

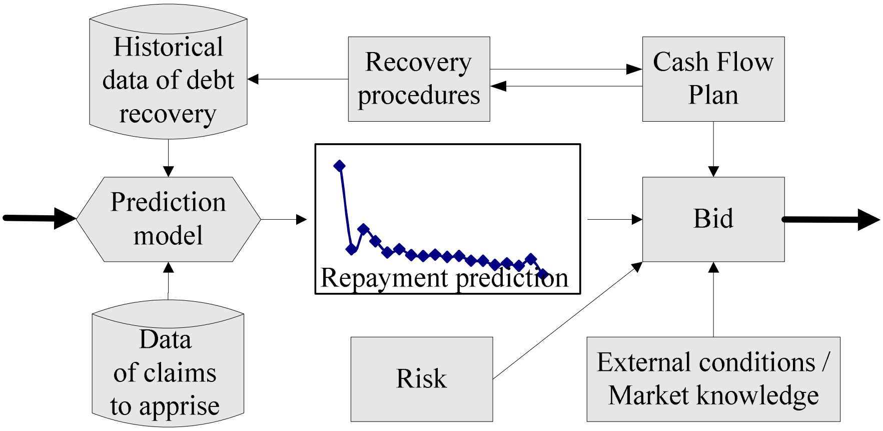

Historical data of debt recovery
Recovery procedures
Cash Flow Plan
Prediction model
Repayment prediction
Bid
Data of claims to apprise
Risk
External conditions / Market knowledge

Fig. 1. The business proces of purchasing a debt portfolio based on repayment prediction

3 Claim Appraisal

3.1 Business Process of Debt Portfolio Recovery

The process of debt portfolio value prediction starts when the first company offers a package of debts and expects a purchase proposal from the second one, see Fig. 1. The second company is usually a specialized debt recovery entity. Based on historical data of debt recovery available for the second company, a prediction model is prepared. The model provides estimation of possible return from the package. The bid is supplemented by additional cost of repayment procedures and cash flow abilities as far as risk and final purchase price are proposed to the first company. The most significant and sensitive part of the process is the repayment value prediction for debt portfolio as there is a strong business need for the method to be designed for efficient and accurate prediction with the time factor.

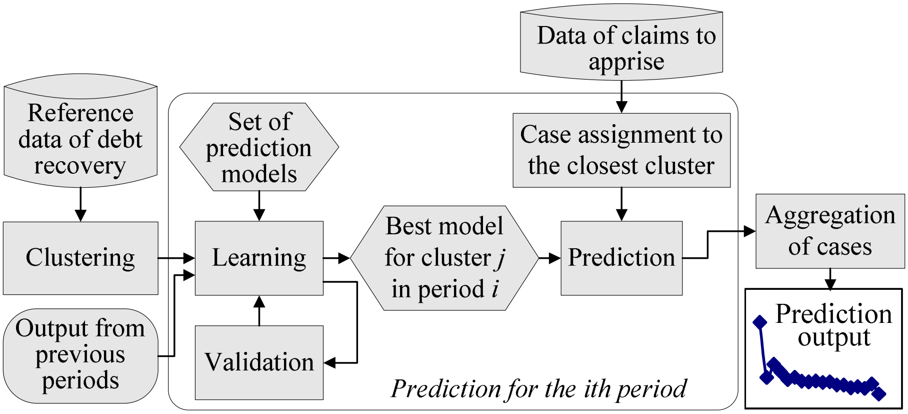

Data of claims to apprise
Reference data of debt recovery
Set of prediction models
Case assignment to the closest cluster
Clustering
Learning
Best model for cluster *j* in period *i*
Prediction
Aggregation of cases
Output from previous periods
Validation
Prediction for the ith period
Prediction output

Fig. 2. Concept of sequential prediction of sequential debt profolio valuation

Having the data of historical claim cases together with their repayment profiles over time, a debt collection company can build a model in order to predict receivables for the new claim set invited for bids. However, in order to be able to evaluate cash flows in the following periods (usually months), the company needs to have possibly precise distribution of the receivables collection. It helps to estimate the final upper value for the considered input debt portfolio. Hence, not only the total aggregated value of the receivables is useful for bidding but also their probable timing, period by period.

3.2 The Concept of the Hybrid Valuation Method

The idea of the hybrid method for prediction of debt recovery value consists of data flows that are executed separately for each period *i* (*M* times), Fig. 2. First, the prepared historical data is clustered into groups of similar debt cases. Next, a set of models is created (learnt) separately for each cluster *j* using the fixed set of common, predefined models. The best one is selected for each cluster and becomes the cluster's predictive model P_{ij}. This assignment is done based on minimization of the standard deviation error. This is the main learning phase followed by the final prediction for the debt portfolio. For each of debt cases, the closest cluster of historical data is determined and the prediction for this case is performed based on the model assigned and trained on that cluster, separately for each period *i*.

The important characteristic of the method is that the predicted value of return on debt in period *i* is taken as the input variable for the *i*+1th period prediction as an additional feature in the model, see Fig. 3.

Historical, reference cases are in general clustered into N^G groups using partitioning method and the best prediction model is separately assigned to each group and each period *i*. Features directly available within the input data set or new ones derived from them are the only used in the clustering process. Besides, clustering is performed for the whole reference set, i.e. for cases being after at least one period of the recovery

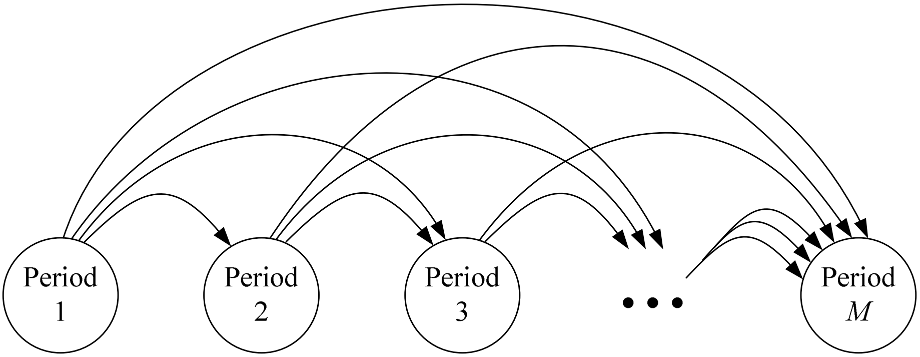


Fig. 3. Input variable dependency in sequential prediction of debt repayment

procedure (period 1). For the following periods, e.g. for period *i*, cases with to short history (being recovered shorter than *i* periods), are just removed from their clusters without re-clustering. As a result, the quantity of each cluster G_{ij} may vary depending on the period *i* and it is smaller for greater *i*. For the *j*th group G_{ij} and the *i*th period, we have: $card(G_{ij}) \geq card(G_{(i+1)j})$. In consequence, there are the same reference groups for all periods but their content decreases for the following periods. This is obvious, because the debt collection company possesses many pending recovery cases, which can be used as references in prediction only for the beginning periods. If the quantity of one cluster for the greater period is too small than this cluster is merged with another, close one for all following periods.

Each group G_{ij} possesses its own representation and the common similarity function is used to evaluate closeness between group G_{ij} and each input case *x* just being predicted. Next, the single closest group, or more precise the assigned model, is applied to the input case *x*.

4 Experimental Setup

For the experimental examination of the proposed method 12 distinct real debt recovery data sets were used. A summary of the data profile is presented in Tab. 1. In total, 20 input features were extracted: 5 continuous, 9 nominal and 6 binary. The experiments were performed using 5 cross-fold validation setup independently applied for each data set [5]. As the proposed prediction process consists of algorithms which efficiency depends on some parameters, some preliminary assessments were applied. The key parameters of the hybrid method are: the number of groups that the clustering process produces, the number and types of predictors used and the method for the selection of the best predictor for each group. The number of groups was adjusted from the range of 5 to 50 by means of X-means algorithm [14]. The average number of groups was 17.6. Three simple predictors were used: M5P tree, logistic regression and regression tree. Decision of taking these relatively simple machine learning approaches was caused by the high computational cost of prediction for each group and each period. The total number of predictions for only one data set was: 17.6 groups * 10 periods * 3 predictors * 5 cross-validations = 2 640 models to be learnt. For 12 data sets, altogether 31 680 predictors were exploited. Obviously, the usage of more complex and sophisticated prediction methods is envisaged in future research. The best predictor assignment to each group is carried out based on the minimization of the prediction standard deviation. The research was implemented and conducted within the R statistical computing environment with the extended and customized algorithms based on RWeka, rJava and tree plug-ins.

In the experiment, the debt recovery value prediction has been conducted for 10 consecutive periods (months). Three different scenarios were realized and finally compared with each other. In each scenario, the output of period *i* is used as the input variable for the following periods. The first scenario assumes simple prediction to be carried out on the single model (regression tree), which is learnt and validated on the training data without clustering. The learning is accomplished separately for each period. The first scenario can be treated as the basic approach for value prediction of sequential and continuous variables. In the second scenario, also without clustering, the assessment of three distinct predictors is performed and the best one is chosen for each period. The full hybrid process is performed in the third scenario, including clustering of the reference data set, see Fig. 3 and 4. Clustered data was used to train all models and the best model was determined for each cluster. Next, in the testing phase, the input cases were assigned to the closest cluster and processed by the assigned predictor. In other words, if an appraisal case is close to a certain cluster, the return value would be predicted by the model assigned to that cluster. The second scenario extends the first one, whereas the third expands the second.

Table 1. Summary of debt recovery data sets

Data set	Number of cases	Data set	Number of cases	Data set	Number of cases	Data set	Number of cases
A	4019	D	3175	G	6818	J	6607
B	3440	E	3736	H	1703	K	2515
C	2764	F	4211	I	4584	L	1104

5 Experimental Results

Having established the methods for debt appraisal, three scenarios were launched and compared with each other in respect of average prediction accuracy. The results of experiments are presented in Tab. 2.

The results of three distinct prediction scenarios revealed that the third scenario (the comprehensive, hybrid approach) performs better by 23% than the first one (basic prediction for sequential, continuous values with the single predictor) and by

Table 2. The results of debt recovery value prediction for three different scenarios

Data set	Prediction accuracy		
	Scenario1: single predictor	Scenario 2: predictor selection	Scenario3: predictor selection with clustering
A	64.56%	67.03%	95.11%
B	53.48%	60.40%	88.26%
C	68.98%	89.51%	93.49%
D	70.35%	80.52%	88.98%
E	73.50%	73.72%	92.20%
F	69.40%	95.56%	95.99%
G	95.59%	96.24%	96.63%
H	79.51%	92.02%	96.11%
I	45.81%	87.97%	89.83%
J	71.33%	86.10%	98.89%
K	68.07%	81.95%	92.14%
L	84.25%	91.22%	93.15%
Average accuracy	70.40%	83.52%	93.40%

10% better than the second (with the best predictor selection). The third final method for debt portfolio valuation stays in high contrast with other simpler approaches, especially as regards the prediction accuracy as well as prediction error stability, see Tab. 4.

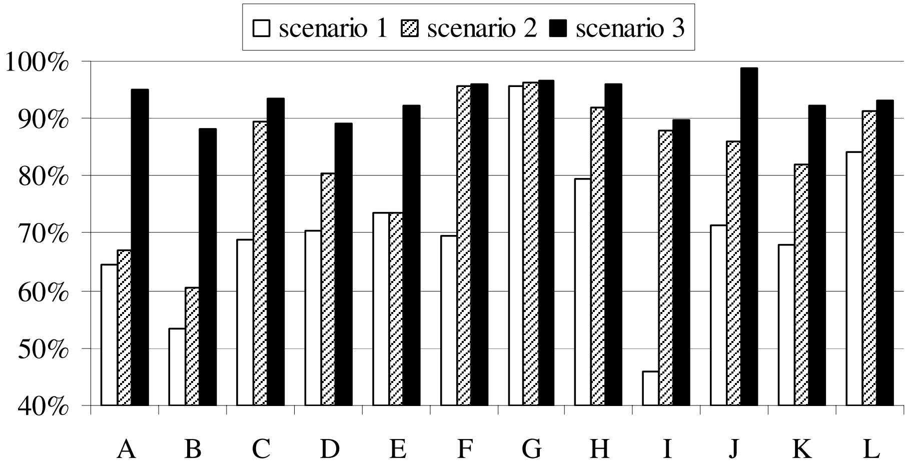


Fig. 4. The accuracy of debt recovery value prediction for three different scenarios for each data set from A to L

Studying ensemble like approaches, it is worth analyzing the error performance in terms of bias and variance factors. The bias and variance reflects the contribution of the prediction error for consecutive periods to the general error prediction [8]. Although, it may happen that the aggregated value of prediction for all periods reveals smaller error rate than the sum of errors from all periods, the prediction for the single period may overestimate or underestimate. Bias and variance of the third scenario prediction is presented in the Fig. 5.

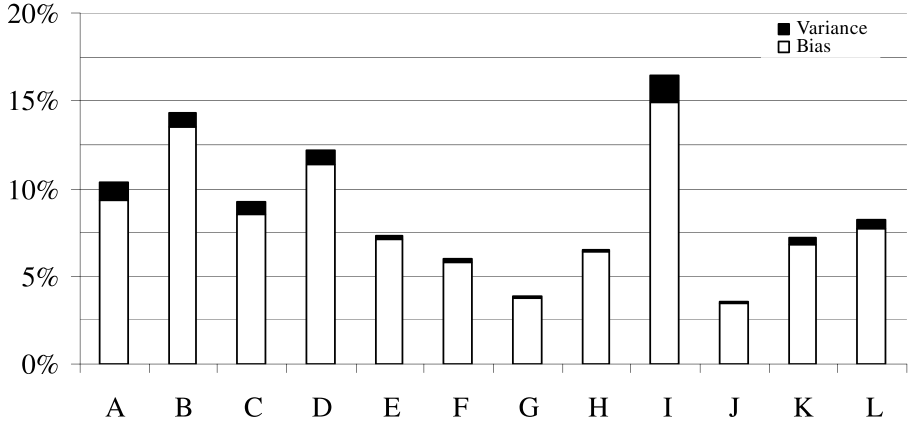


Fig. 5. Bias / variance decomposition for sequential (period) prediction error

As seen in Fig. 5, the stability with respect to prediction error over time is directly reflected in low variance term, concerning most of the error in the bias.

6 Conclusions and Future Work

In order to predict debt portfolio value, the proper hybrid method has been suggested and examined on real data. The experimental results support the conclusion that combined prediction solutions are more accurate and may be efficiently applied to debt recovery valuation.

In the future studies, many further aspects improving the method will be considered, in particular: combination of distinct types of classifiers, models' tuning using genetic based optimization [13] and adaptive clustering.

The application of the similar hybrid concept is also considered to be applied to social-based recommender systems [10].

Acknowledgments. The work was supported by The Polish Ministry of Science and Higher Education, the development project, 2009-11.

References

1. Aburto, L., Weber, R.: A Sequential Hybrid Forecasting System for Demand Prediction. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 518–532. Springer, Heidelberg (2007)
2. Ali, S., Smith, K.: On learning algorithm selection for classification. Applied Soft Computing 6(2), 119–138 (2006)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
4. Chou, C.-H., Lin, C.-C., Liu, Y.-H., Chang, F.: A prototype classification method and its use in a hybrid solution for multiclass pattern recognition. Pattern Recognition 39(4), 624–634 (2006)
5. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10(7), 1895–1923 (1998)
6. Eastwood, M., Gabrys, B.: Building Combined Classifiers, A chapter in Knowledge Processing and Reasoning for Information Society. In: Nguyen, N.T., Kolaczek, G., Gabrys, B. (eds.), pp. 139–163. EXIT Publishing House, Warsaw (2008)
7. Gabrys, B., Ruta, D.: Genetic algorithms in classifier fusion. Applied Soft Computing 6(4), 337–347 (2006)
8. Garcia-Pedrajas, N., Ortiz-Boyer, D.: Boosting k-nearest neighbor classifier by means of input space projection. Expert Systems with Applications 36, 10570–10582 (2009)
9. Kajdanowicz, T., Kazienko, P.: Hybrid Repayment Prediction for Debt Portfolio. In: ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 850–857. Springer, Heidelberg (2009)
10. Kazienko, P., Musiał, K., Kajdanowicz, T.: Multidimensional Social Network and Its Application to the Social Recommender System. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (in press, 2009)
11. Keles, A., Kolcak, M., Keles, A.: The adaptive neuro-fuzzy model for forecasting the domestic debt. Knowledge-Based Systems 21(8), 951–957 (2008)
12. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. John Wiley & Sons, Inc., Chichester (2004)
13. Lin, P.-C., Chen, J.-S.: A genetic-based hybrid approach to corporate failure prediction. International Journal of Electronic Finance 2(2), 241–255 (2008)
14. Pelleg, D., Moore, A.W.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann Publishers Inc., San Francisco (2000)
15. Ravi, V., Kurniawan, H., Nwee, P., Kumar, R.: Soft computing system for bank performance prediction. Applied Soft Computing 8(1), 305–315 (2008)
16. Rud, O.: Data Mining Cookbook. Modeling Data for Marketing, Risk, and Customer Relationship Management. John Wiley & Sons, Inc., Chichester (2001)
17. Swanson, N.R., White, H.: A model selection approach to assessing the information in the term structure using linear model and the artificial neural network. Journal of Business and Economics Statistics 13, 265–275 (1995)
18. Zurada, J., Lonial, S.: Comparison of The Performance of Several Data Mining Methods For Bad Debt Recovery In The Healthcare Industry. The Journal of Applied Business Research 21(2), 37–53 (2005)

A Computer-Assisted Colorization Approach Based on Efficient Belief Propagation and Graph Matching

Alexandre Noma[1], Luiz Velho[2], and Roberto M. Cesar-Jr[1,⋆]

[1] IME-USP, University of São Paulo, Brazil
alex.noma@gmail.com, cesar@ime.usp.br
[2] IMPA, Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil
lvelho@impa.br

Abstract. Region-based approaches have been proposed to computer-assisted colorization problem, typically using shape similarity and topology relations between regions. Given a colored frame, the objective is to automatically colorize consecutive frames, minimizing the user effort to colorize the remaining regions. We propose a new colorization algorithm based on graph matching, using Belief Propagation to explore the spatial relations between sites through Markov Random Fields. Each frame is represented by a graph with each region being associated to a vertex. A colored frame is chosen as a 'model' and the colors are propagated to uncolored frames by computing a correspondence between regions, exploring the spatial relations between vertices, considering three types of information: adjacency, distance and orientation. Experiments are shown in order to demonstrate the importance of the spatial relations when comparing two graphs with strong deformations and with 'topological' differences.

1 Introduction

Computer-Assisted Cartoon Animation is one of the most challeging areas in the field of Computer Animation. Since the seminal paper of Ed Catmull [4], which describes the 2D animation pipelines and its main problems, intense research has been carried out in this area. Nonetheless, many problems remain unsolved due to several difficulties. On one hand, traditional cartoon animation relies on artistic interpretation of reality. On the other hand, 2D animation relies on the dynamics of an imaginary 3D world, depicted by 2D strokes from the animator. These aspects make most of the tasks in computer-assisted cartoon animation very hard inverse problems, which are ill-posed and closely related to perceptual issues.

Approximations to such problems are required and we may take advantage of structural pattern recognition, which differs from the statistical approach

[⋆] The authors are grateful to FAPESP, CNPq, CAPES and FINEP for financial support, and to Cesar Coelho for the animations used in the experiments.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 345–352, 2009.

because, besides the appearance features, the former explores structural relations between the patterns in order to improve the classification. One of the most important ways to implement structural pattern recognition methods relies on graph representation and matching. Here, we focus on matching two graphs, called model and input graphs. The model graph contains all classes, while the input graph represents the patterns to be classified.

Markov Random Fields (MRFs) have been successfully applied to low level vision problems such as stereo and image restoration. In order to obtain a solution, a cost (or energy) function must be minimized, consisting of the observation and the Markov components [2,7]. The observation component evaluates the appearance (e.g. gray levels of pixels), and the Markov component priviledges particular configurations of labels, simplified to pairwise interactions between sites (e.g. smoothness). Currently, there are two popular approaches to estimate a solution for MRFs: Graph Cuts (GC) [2] and Belief Propagation (BP) [7]. GC are based on an efficient implementation of min-cut / max-flow and BP on a message passing approach in order to propagate appearance and smoothness information through the graph. While the GC based methods are restricted to Markov components representing semi-metrics, the BP based approaches are more general since they do not impose any explicit restriction to the cost function.

A general approach based on graph matching, MRF and BP, with pairwise interactions between sites, is proposed in the present paper for point matching problems. Here, this framework is applied to computer-assisted colorization for cartoon animation [1]. Given an animation sequence, the goal is to track the 2D structural elements (representing regions) throughout the sequence to establish correspondences between the regions from consecutive frames [1] in order to automatically propagate the colors to different frames. This allows fast modifications on the colors of entire animation sequences by simply editing the rendering of one single frame. In this case, each region is represented by its centroid and we want to find a correspondence between two points sets, one from the colored and other from the uncolored frame.

The matching between regions corresponds to a 'weaker' form of isomorphism. In practice, the isomorphism is too restrictive and a weaker form of matching is the subgraph isomorphism, which requires that an isomorphism holds between one of the two graphs and a vertex-induced subgraph of the other. An even weaker form of isomorphism is the maximum common subgraph (MCS), which maps a subgraph of the first graph to an isomorphic subgraph of the second one [6].

Closely related to this work is the one due to Caelli and Caetano [3]. They proposed three methods for graph matching based on MRFs, evaluated through artificial experiments to match straight line segments. A key point which has not been explored is the importance that spatial relations can represent, specially when the simplest case of (pairwise) interactions between sites is considered. Here, we extend the efficient BP message computation described in [7], keeping efficiency while exploring three types of structural information simultaneously: adjacency, distance and orientation between patterns. This strategy makes the

Markov component much more discriminative than just 'smoothness'. (Note that in our case, smoothness is also explored by the adjacency between patterns.)

A previous work for computer-assisted colorization was presented in [1], which was based on three factors: region area, point locations and the concept of Degree of Topological Differences (DTD) in order to explore the adjacencies between regions. Here, we propose a simple approach based on area, contour length and point locations used to encode the spatial relations as described in [5] and [8]. The objetive of the proposed general framework based on MRF is to overcome the main difficulty in graph matching problems, expressed by the following question. *How to match two 'topologically' different graphs, with different sizes (different number of vertices and edges), and possibly with 'strong deformations' in terms of appearance / structure between the corresponding patterns?* The proposed method attempts to answer this question by exploring the contextual information, given by the 'labeled' neighbors, through MRFs.

This paper is organized as follows. In Section 2, we formulate the generic graph matching problem as MRF. Section 3 describes the proposed probabilistic optimization approach based on BP. In Section 4, there is a description of our proposed solution to the colorization problem. Section 5 is dedicated to the experimental results. Finally, some conclusions are drawn in Section 6.

2 Graph Matching as MRFs

An Attributed Relational Graph (ARG) $G = (V, E, \mu, \nu)$ is a directed graph where V is the set of vertices of G and $E \subseteq V \times V$ the set of edges. Two vertices $p \in V$, $q \in V$ are adjacent if $(p, q) \in E$. μ assigns an attribute vector to each vertex of V. Similarly, ν assigns an attribute vector to each edge of E. Following the same notation used in [5], we focus on matching two graphs, an input graph G_i, representing the scene (input image) with all patterns to be classified, and a model graph G_m, representing the template with all classes. Given two ARGs, $G_i = (V_i, E_i, \mu_i, \nu_i)$ and $G_m = (V_m, E_m, \mu_m, \nu_m)$, we define a MRF on the input graph G_i. For each input vertex $p \in V_i$, we want to associate a model vertex $\alpha \in V_m$, and the quality of a mapping (or labeling) $f : V_i \to V_m$ is given by the cost function defined in Equation 1, which must be minimized.

$$E(f) = \sum_{p \in V_i} D_p(f_p) + \lambda_1 \sum_{(p,q) \in E_i} M(f_p, f_q) , \tag{1}$$

where λ_1 is a parameter to weight the influence of the Markov component on the result. Each vertex in each ARG has an attribute vector $\mu_i(p)$ in G_i and $\mu_m(\alpha)$ in G_m. The observation component $D_p(f_p)$ compares $\mu_i(p)$ with $\mu_m(f_p)$, assigning a cost which is proportional to the vertices attributes dissimilarity. Each directed edge in each graph has an attribute vector $\nu_i(p, q)$ in G_i and $\nu_m(\alpha, \beta)$ in G_m, where $(p, q) \in E_i$ and $(\alpha, \beta) \in E_m$. The Markov component $M(f_p, f_q)$ compares $\nu_i(p, q)$ and $\nu_m(f_p, f_q)$, assigning a cost which is proportional to the edges attributes dissimilarity.

3 Optimization Based on BP

In order to find a labeling with minimum cost, we use the max-product BP [7], which works by passing messages around the graph according to the connectivity given by the edges. Each message is a vector whose dimension is given by the number of possible labels $|V_m|$. Let m^t_{pq} be the message that vertex p sends to a neighbor q at iteration t. Initially, all entries in m^0_{pq} are zero and, at each iteration, new messages are computed as defined by Equation 2.

$$m^t_{pq}(f_q) = \min_{f_p}\left(M(f_p, f_q) + D_p(f_p) + \sum_{s\in\mathcal{N}_p\setminus\{q\}} m^{t-1}_{sp}(f_p)\right) \tag{2}$$

where $\mathcal{N}_p \setminus \{q\}$ denotes the neighbors of p except q. After T iterations, a belief vector is computed for each vertex:

$$b_q(f_q) = D_q(f_q) + \sum_{p\in\mathcal{N}_q} m^t_{pq}(f_q)\ . \tag{3}$$

Finally, the label f^*_q which minimizes $b_q(f_q)$ individually at each vertex is selected.

In the following, we describe an efficient computation of each vector message. Equation 2 can be rewritten as [7]:

$$m^t_{pq}(f_q) = \min_{f_p}\left(M(f_p, f_q) + h(f_p)\right)\ , \tag{4}$$

where $h(f_p) = D_p(f_p) + \sum m^{t-1}_{sp}(f_p)$. In order to compute the messages efficiently, based on the Potts model [7], we assume:

$$m^t_{pq}(f_q) = \min\left(H(f_q), \min_{f_p} h(f_p) + d\right)\ . \tag{5}$$

The main difference explored in the present paper relies on $H(f_q)$, which takes into account the edges of the model graph:

$$H(f_q) = \min_{f_p\in\mathcal{N}_{f_q}\cup\{f_q\}}\left(h(f_p) + M(f_p, f_q)\right)\ , \tag{6}$$

where, besides the neighbors in $\mathcal{N}_{f_q}$, it is also necessary to examine the possibility that p and q have the same label.

Thus, to compute each message vector, the amortized time complexity can be upper bounded by the number of edges in the model graph. The standard way to compute a single vector message update is to explicitly minimize Equation 2 over f_p for each choice of f_q, which is quadratic on the number of labels. We propose a modification of the orginal algorithm in [7] to compute each message in linear time on $|E_m|$, which is based on using Equation 6 instead of $H(f_q) = h(f_q)$ for the Potts model described in [7].

4 Computer-Assisted Colorization

The proposed general framework was applied to the computer-assisted colorization problem. Given an animation sequence, the 2D structural elements (representing regions) must be tracked throughout the sequence in order to establish correspondences between the regions from consecutive frames. The goal is to automatically propagate the colors to different frames using this correspondence, which is obtained by a MCS through a cost function, considering two types of information, appearance and structure.

More specifically, given two frames, one colored and the other uncolored, we want to find a correspondence between regions from both frames in order to propagate the colors from the colored to the uncolored frame. Each frame is represented by an ARG. The colored one is the model ARG, while the uncolored one is the input ARG.

Both input and model graphs are obtained similarly. Let W be a set of regions (connected components) defined by the animator strokes in the drawing. Each region in W is represented by its centroid, which is represented by a vertex. Edges are created between adjacent regions, assuming that important contexts are given by adjacent neighbors. Both input and model consist of planar graphs, each one having $|E| = O(|V|)$ edges. Therefore, the algorithm to compute each message vector, described in Section 3, is linear on the number of labels $|V_m|$.

The appearance information is represented by the vertex attributes and the structure by the edge attributes. After T iterations, the BP approach computes the belief vector for each vertex, representing the costs of each label, and assigns a label with minimum cost to obtain a homomorphism. In order to obtain a MCS and to guarantee that the mapping is bijective between subsets of V_i and V_m, we applied the same post-processing as described in [8]: for each model vertex, we kept the cheapest input vertex, and the remaining input vertices were associated to a NULL label, indicating they are not classified, leaving ambiguous cases for the animator to decide which color must be used to the unclassified (uncolored) regions.

Next we describe each term of the energy function in Equation 1.

4.1 Observation Component

For each vertex v, the appearance information $\mu(v)$ consists of two attributes: the area and the contour length of the region corresponding to vertex v. For the observation component, we used

$$D_p(f_p) = \max\left\{\frac{|\mu_A(p) - \mu_A(f_p)|}{\mu_A(p)}, \frac{|\mu_C(p) - \mu_C(f_p)|}{\mu_C(p)}\right\}, \tag{7}$$

where μ_A and μ_C represents the area and the curve length, respectively, $p \in V_i$ and $f_p \in V_m$. In order to map p to f_p, both attributes must match simultaneously, thus leaving ambiguous regions to the user.

4.2 Markov Component

For each directed edge $e \in E$, a single edge attribute $\nu(e)$ is defined as the (normalized) vector corresponding to the directed edge. The Markov component compares edge attributes through the dissimilarity function defined by Equation 8 [5], which compares pairs of vectors in terms of angle and lengths in order to characterize the spatial relations.

$$c_E(\boldsymbol{v_1}, \boldsymbol{v_2}) = \lambda_2 \frac{|\cos\theta - 1|}{2} + (1 - \lambda_2)\big||\boldsymbol{v_1}| - |\boldsymbol{v_2}|\big| \ , \tag{8}$$

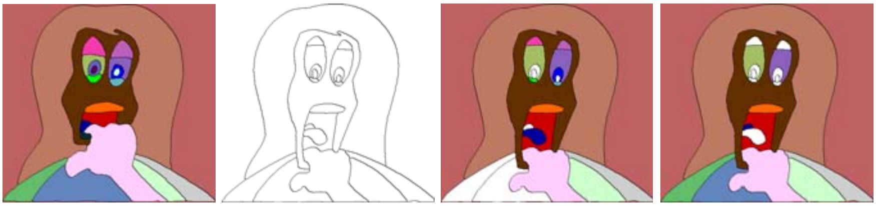
where θ is the angle between the two vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, $|.|$ denotes the absolute value, $|\boldsymbol{v}|$ denotes the length of $\boldsymbol{v}$ (assuming all lengths $|\boldsymbol{v}|$ are normalized between 0 and 1), and λ_2 is a parameter to weight the importance between the two terms. The Markov component $M(f_p, f_q)$ is defined as the edges dissimilarities described in [5]:

$$M(f_p, f_q) = \begin{cases} c_E\big(\nu_i(p,q), \nu_m(f_p, f_q)\big), \text{ if } (f_p, f_q) \in E_m \\ d, \quad \text{if } (f_p, f_q) \notin E_m \quad \text{and} \quad f_p \neq f_q \end{cases} \tag{9}$$

where the first case compares the respective vectors using Equation 8, and the second penalizes the cost with a positive constant d, encouraging adjacent vertices to have the same label. In this case, $M(f_p, f_q) = M(\alpha, \alpha) = c_E\big(\nu(p,q), \mathbf{0}\big) < d$, proportional to $|\nu(p,q)|$ (because it is assumed $\theta = 0$ in this case), thus penalizing distant vertices. This fact implies that the proposed Markov component is not a semi-metric, since $M(\alpha, \alpha)$ may be different from zero, and the GC [2] based methods are not guaranteed to produce good approximations. Fortunately, this limitation does not apply for the BP algorithm.

5 Experimental Results

The proposed method was tested on four animations: 'cufa', 'calango', 'wolf' and 'face'. We tested three factors: deformations in appearance due to large



(model) (input) (our result) (previous work)

Fig. 1. 'Face' example. From the colored frame (model), we want to colorize the next uncolored input frame. The results from our method and from a previous work. For all colorization experiments, we used $\lambda_2 = 0.5$ (Equation 8), penalty $d = 1.0$ (Equation 9), and all vectors were normalized by the maximum length of all edge attributes (vectors).

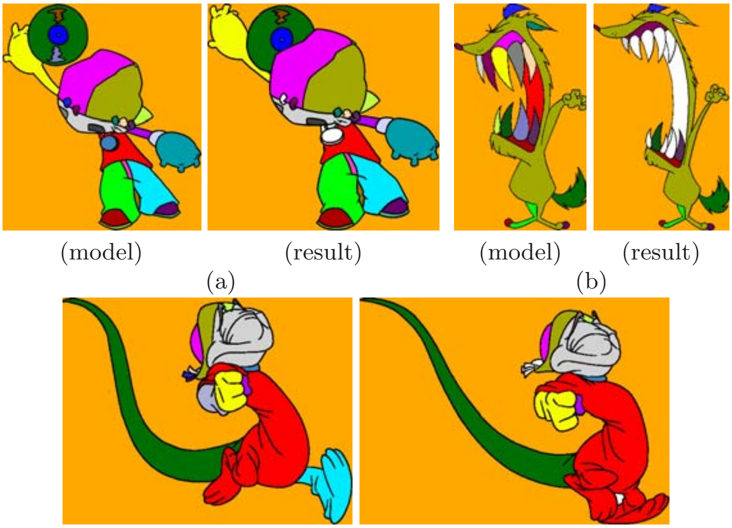
Table 1. Quantitative results from the tested animations

animation	frames	$\lvert V_m \rvert$	$\lvert V_i \rvert$	# wrong colorizations	# missings	# correctly unclassified
cufa	42, 43	35	36	0	4 (11.11%)	2 (100.0%)
calango	24, 25	23	21	1 (4.76%)	3 (14.28%)	1 (100.0%)
wolf	26, 27	23	23	1 (4.35%)	8 (34.78%)	2 (100.0%)
face	–	18	22	0	5 (22.72%)	1 (100.0%)

variations in the region area / contour, deformations in structure due to large motions, and 'topological' differences caused by merging / splitting of regions.

Figures 1 and 2 illustrate the experiments. In all examples, one column illustrates the colored frame for the model and another showing the colorization result. White regions represent the unclassified or uncolored regions.

In Figure 2(a), there is a partial occlusion of the disc, causing one region to disappear and a large change in the 'area'. Also, some regions appear due to splitting of the region near the necklace and medal. Figure 2(b) illustrates a challenging example, with strong deformations on both appearance and structure, merging (e.g. gingiva) and splitting (background) of regions, and new teeths, causing great topological incompatibilities. In Figure 2(c), some regions disappear due to merging of regions (leg and arm).



(model) (result) (model) (result)

(a) (b)

(model) (result)

(c)

Fig. 2. Example of colorization on the (a) 'cufa', (b) 'wolf' and (c) 'calango' animations. For each example, we present the model and the corresponding result, respectively.

Figure 1 is used to illustrate a comparison against [1]. Quantitative results are shown in Table 1. For instance, in the 'face' example, all the colored regions were correctly matched by our approach (no wrong colorization). Although there were 5 missing regions produced by our approach (and 1 new region correctly unclassified, above the tonge), the method in [1] produced 8 missings (an improvement of 37.5%). Among the 5 missings, 4 were due to changes in the 'adjacency' property, penalized by our approach: two regions of the body and the two pupils. $|V_m|$ and $|V_i|$ denotes de number of model and input vertices, respectively.

6 Conclusions

This paper has proposed a novel general framework for graph matching, using spatial relations through Markov Random Fields (MRFs) and efficient belief propagation (BP) for inference. The edges dissimilarities described in [5] were used as a Markov component, leading to a very useful tool for point matching problems. The key to achieve efficiency was the assumption that important contextual information is concentrated on close neighbors. Both input and model patterns were represented by planar graphs, allowing an efficient algorithm to compute the messages, i.e. linear on the number of labels.

For the computer-assisted colorization problem, we have shown encouraging results, illustrating the benefits of our approach for large deformations in appearance and structure, and for topological incompatibilities on the two graphs being matched, induced by merging and splitting of regions.

Future works include the aplication of the proposed method to other important vision problems, such as image segmentation and shape matching.

References

1. Bezerra, H., Feijo, B., Velho, L.: A Computer-Assisted Colorization Algorithm based on Topological Difference. In: 19th SIBGRAPI, pp. 71–77 (2006)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. PAMI 23(11), 1222–1239 (2001)
3. Caelli, T., Caetano, T.: Graphical models for graph matching: approximate models and optimal algorithms. PRL 26(3), 339–346 (2005)
4. Catmull, E.: The problems of computer-assisted animation. SIGGRAPH 12(3), 348–353 (1978)
5. Consularo, L.A., Cesar-Jr, R.M., Bloch, I.: Structural Image Segmentation with Interactive Model Generation. In: ICIP, vol. 6, pp. 45–48 (2007)
6. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty Years Of Graph Matching In Pattern Recognition. IJPRAI 18(3), 265–298 (2004)
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Belief Propagation for Early Vision. IJCV 70(1), 41–54 (2006)
8. Noma, A., Pardo, A., Cesar-Jr, R.M.: Structural Matching of 2D Electrophoresis Gels using Graph Models. In: 21st SIBGRAPI, pp. 71–78 (2008)

Signal Analysis for Assessment and Prediction of the Artificial Habitat in Shrimp Aquaculture

José Juan Carbajal Hernández, Luis Pastor Sanchez Fernandez,
José Luis Oropeza Rodríguez, and Edgardo Manuel Felipe Riverón

Centre of Computer Research – National Polytechnic Institute, Av. Juan de Dios Bátiz, Col. Nueva. Industrial Vallejo, México D.F., México
juancarvajal@sagitario.cic.ipn.mx,
{lsanchez,joropeza,edgardo}@cic.ipn.mx

Abstract. This paper presents a novel work for prediction of artificial habitat in shrimp aquaculture based on environmental signal analysis. The physical-chemical variables that are involved into the system are studied for modeling and predicting environmental patterns. The prediction model is built using AR models that reconstruct a partial section of a particular measured signal. The physical-chemical variables are classified based on the negative ecological impact using a new statistical model that calculates the frequency and the deviation of the measurements. A fuzzy inference system processes the level classifications using aquaculture rules that define all the cases calculating the condition of the shrimp habitat.

Keywords: fuzzy inference systems, prediction, signal analysis, Assessment.

1 Introduction

The main purpose on water management and aquaculture systems is to control and maintain the optimal conditions for the surviving and growing of the organisms in normal farming conditions [1]. The early detection of potential problems can be decisive in the organism health and the economical activity of the farm. The negative impact of a set of physical-chemical variables can be assessed when they occur, but estimate the future condition of the ecosystem can be a difficult task since they are not tools for solving this problem. In other hand, if the concentrations levels are predicted, potential danger situations could be avoided before they appear [2]. There is a lack of methodologies for prediction and assessment of water quality; the methods actually developed have several weaknesses were the lack of a reasoning process in the assessment of the information is the main problem [3], [4], [5], in addition, the prediction process usually is confused as a present condition to be predominant for the rest of the day [6].

2 Data Collection

A set of physical-chemical variables compounds the ecosystem of the shrimp; this set must be under control and in optimal ranges. As a result of this condition, the features

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 353–360, 2009.

of the variables are studied with the objective to determine the frequency and importance of their behaviors [2].

The measurements of the variables depend of the exactitude of how a supervisor monitors the aquaculture system. A complete farm was monitored in Rancho Chapo located in Sonora, Mexico. The higher impact variables measured were temperature, dissolved oxygen, salinity and pH, using a sensor device for each variable. The period of monitoring was of 15 minutes. The data set contains four months of measurements; it means a register of 9312 values per variable. The classification levels of the physical-chemical variables (status) are defined in Table 1, for dissolved oxygen we chosen "hypoxia", "low" and "normal", for the temperature and salinity variables we chosen "low", "normal" and "high", and for the pH variable we chosen "acid", "low", "normal", "high", and "alkaline".

Table 1. Classification levels, tolerances (Tol) and limits (Lim) of physical-chemical variables

Variables	Hypoxia Acid	Low	Normal	High	Alkaline	Tol.	Lim.
Temp (*°C*)	-------	0 – 23	23 - 30	30 - ∞	-------	±1	±1
Sal (mg/L*)*	-------	0 – 15	15 - 25	25 - ∞	-------	±1	±1
DO (mg/L)	0 – 3	3 – 6	6 - 10	10 - ∞	-------	±0.5	±0.5
PH	0 – 4	4 – 7	7 – 9	9 - ∞	10 - 11	±0.5	±0.5

3 Series Prediction

3.1 Preprocessing

Smoothing

The variables signals have several peaks values, this behavior can be generated due a failed device, human error or environmental situations. The four signals of the physical-chemical variables are smoothed in order to be more easily for processing. A moving average weighted filter works using an average of signal points (measured concentrations) for producing new output points of the new filtered signal and smoothing it [7]. The smoothing process of the physical-chemical variables can be calculated as follows:

$$y(n) = \sum_{i=0}^{N} b_i x(n-i) \tag{1}$$

where *x(n)* is the original signal, *y(n)* is the new output signal, *N* is known as the filter order, b_i are the Spencer 15 terms coefficients defined as $\frac{1}{320}$ [-3, -6, -5, 3, 21, 46, 67, 74, 67, 46, 21, 3,-5,-6,-3]. The smoothing process using a moving average weighted filter is:

$$y(n) = -\frac{3}{320}x(n) - \frac{6}{320}x(n-1) - \frac{5}{320}x(n-2) + \cdots - \frac{3}{320}x(n-14) \tag{2}$$

The Fig. 1 shows examples of the original and smoothed measured variables, where the random perturbations are suppressed.

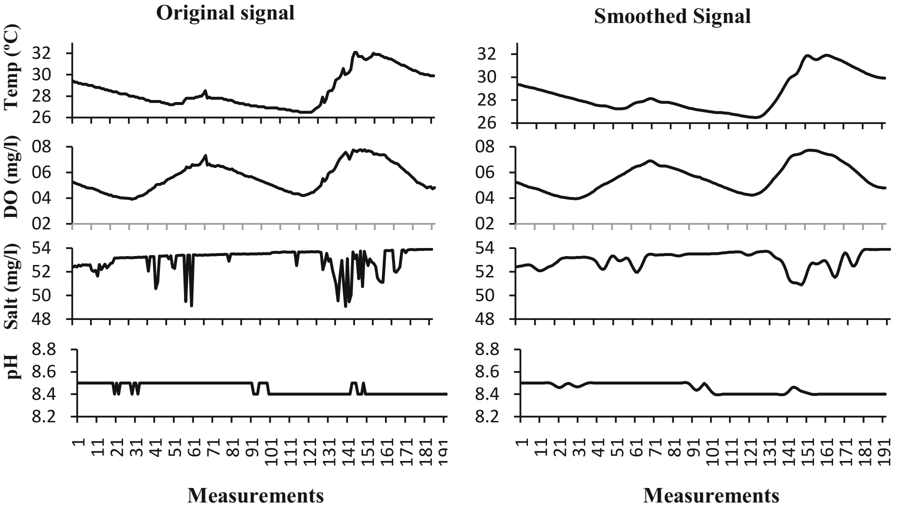


Fig. 1. Original and smoothed signal of the physical-chemical variables using a moving average filter

Detrending

The environmental series usually contain some constant amplitude offset components or trends. The amplitudes of these trends sometimes corrupt the results of series modeling. Therefore, it is needed to remove them before performing further analysis [8]. The trend is calculated using the linear regression method, where the equation of the estimated trend for a particular variable can be expressed as follows:

$$y = a_0 + a_1 x + E \tag{3}$$

where a_0 y a_1 are coefficients that represent the intersection with the abscise axis and the pendent respectively, y is the physical-chemical variable (temp, salt, DO and pH) and E is the error between the modeled and the observed values. The coefficient a_1 can be calculated using:

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \tag{4}$$

where n is the number of points of the series, and x_i is the ith measurement. For the a_0 coefficient:

$$a_0 = \bar{y} - a_1 \bar{x} \tag{5}$$

3.2 Autoregressive Model (AR)

The AR model of a series allows predicting the current value x_t, based on past values $x_{t-1}, x_{t-2}, \ldots, x_{t-n}$ and a prediction error. The n parameter determines the number of past values that are used for predicting the current value (model order). The model order can be estimated using an error estimator, it is known as Akaike criterion [9]:

$$AIC = E_n\left(1 + \frac{2n}{L}\right) \tag{6}$$

where L is the number of points in the time series, n is the model order and E_n is the prediction error. The AR models that describe the physical-chemical variables using the model order estimated with the AIC are:

$$temp_t = \sum_{i=1}^{79} a_i temp_{t-i} + e_t \tag{7}$$

$$DO_t = \sum_{i=1}^{80} a_i DO_{t-i} + e_t \tag{8}$$

$$Salt_t = \sum_{i=1}^{20} a_i Salt_{t-i} + e_t \tag{9}$$

$$pH_t = \sum_{i=1}^{56} a_i pH_{t-i} + e_t \tag{10}$$

where a_i is the i^{th} AR coefficient, e_t is the predicted error and p is the model order.

The Fig. 2 shows the reconstruction of the physical-chemical signals, where a total of 96 points (one day/24 hours) where predicted using the AR(p) model suggested.

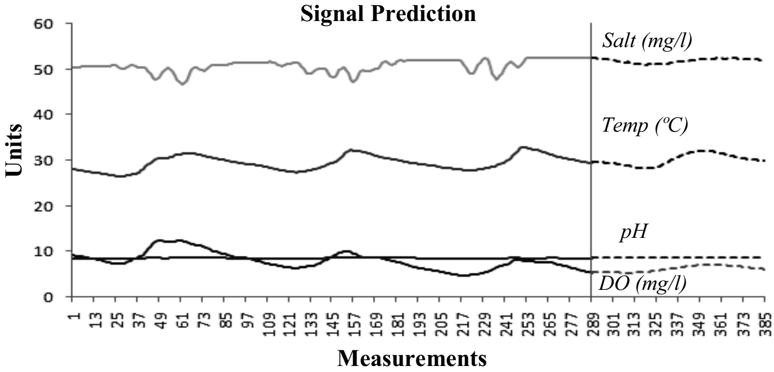


Fig. 2. Prediction of the physical-chemical variables. The AR model predicts 96 measurements (24 hours).

4 Assessment

Historical Water Quality Index

The **H**istorical **W**ater **Q**uality **I**ndex (HWQI) asses aquaculture shrimp systems using historical information (variables set measurements). The result of the assessment is a status given by the behavior of the variables measured. The levels of classification of the HWQI are defined as *Excellent, Good, Regular* and *Poor*.

The HWQI works in two phases; first it calculates the physical-chemical index, which classifies the effect level of a variable value in the ecosystem. The second phase consists on evaluate the result information of the Γ index by a reasoning process using a fuzzy inference system.

Physical – chemical assessment index (Γ)

The fuzzyfication process is done using the physical-chemical index (Γ) and it comprises three factors; frequency, amplitude and deviation of the failed tests [5]. The frequency (α) of failed tests in a set of 96 measurements is calculated using the next expression:

$$\alpha = \frac{\mathrm{m_f}}{\mathrm{m_T}} = \frac{\mathrm{m_f}}{96} \tag{11}$$

where m_T is the total number of measurements and m_f is the number of measurements out of the desired range. The average of the number of deviations (β) of failed test is calculated using:

$$\beta = \frac{\sum_i^n e_i}{m_T} = \frac{\sum_{i=1}^{96} e_i}{96} \tag{12}$$

where *e* is the deviation of the i^{th} failed measurement (out of the desired level), *i*: 1, 2, … n, *n* is the number of the deviations and m_T is the number of total tests. The variable *e* can be expressed as follows:

when the value must not fall below the level:

$$e = \frac{l_b - t_b}{l_b \ - m} \tag{13}$$

when the value must not exceed the level:

$$e = \frac{m - l_a}{t_a - \ l_a} \tag{14}$$

where *m* is the value of the test, *la* is the upper limit of the range to evaluate, *ta* is the upper tolerance, *lb* is the lower limit of the evaluated range and *tb* is the lower tolerance of the range (Table 1).

The Physical – chemical index can be expressed as follows:

$$\Gamma \ = \sqrt{\frac{\left(\frac{\mathrm{m_f}}{96}\right)^2 + \left(\frac{\sum_i^n e_i}{96}\right)^2}{2}} \tag{15}$$

Finally the membership input equation for the fuzzy inference system is defined as follows:

$$\mu_{variable} = \begin{cases} 0 & 1 < \Gamma \\ 1 - \Gamma & 0 \leq \Gamma \leq 1 \end{cases} \tag{16}$$

Reasoning process

The level classifications of the particular variables are processed using a set of rules that involves all the cases of the habitat condition. There are some expressions that are frequently used by experts in water management, these kinds of expressions construct the fuzzy language of the FIS, and these rules can be expressed as follows:

Rule 1: If *Temp* is normal and *Salt* is normal and *pH* is normal y *DO* is normal then *WQI* is Excellent

Rule 2: If *Temp* is normal and *Salt* is normal and *pH* is normal y *DO* is low then *WQI* is Good

The size of the rule set depends of the number of rules that are involved in the environment; a total of 135 rules have been used in this case.

Predicted Water Quality Index (PWQI)

The reasoning process generates one output membership by one rule, and this result is used to establish the membership of the output function (Fig. 3).

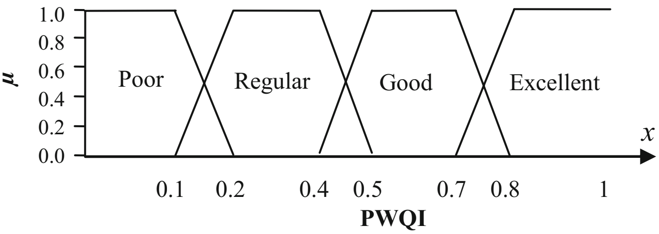


Fig. 3. Membership functions for PWQI

For transforming the final results (the indices results processed by the reasoning process) in a real output value (condition of the shrimp habitat), it is needed to realize the aggregation process, where all the resulting membership functions generate one final function ($\mu_{out}(x)$). The deffuzification process is done when the center of gravity is calculated using $\mu_{out}(x)$, to do this the centroid method is used as a deffuzification solution:

$$PWQI = \frac{\int x\mu_{out}(x)dx}{\int \mu_{out}(x)dx} \tag{17}$$

The final score for the PWQI index have a range from the center of the *poor* function to the center of the *excellent* function [0.078, 0.87], therefore the different status values are located inside this range; *bad* is 0.078, *regular* is 0.3, *good* is 0.6 and *excellent* is 0.87.

5 Results

A predictability analysis shows the perform of the system. Prediction tests were done using one and two days of information. The PWQI (predicted day) results were compared with the HWQI (evaluated day) showing some interesting remarks (Fig. 4). The left column shows the predicted values versus the real values, where the curves have similar results. A second prediction analysis is showed in right column, where a relationship between estimated and real values was done, the nearest points to the diagonal line are a closer prediction to the real assessment of the water quality. The results show a closer relationship between HWQI and PWQI indices.

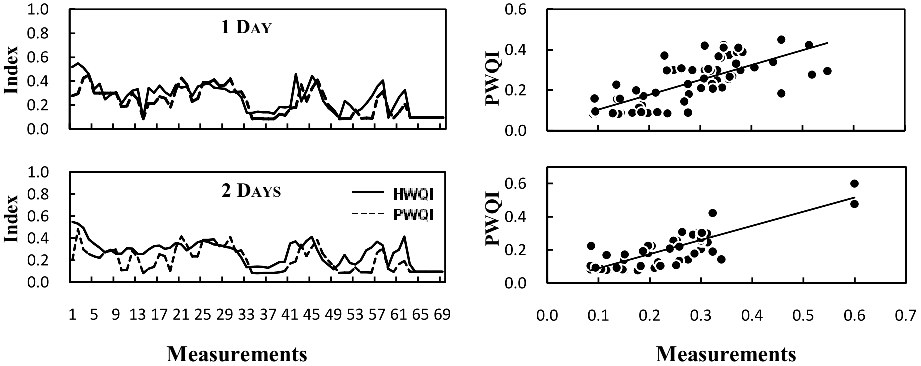


Fig. 4. Results of the prediction and assessment of the artificial habitat. The left column shows a comparison with different prediction horizons (24 and 48 hours) between the real and predicted data. The right column shows the relationship of predicted (PWQI) and real values (HWQI), where the diagonal line represents the exactitude of the prediction.

When the number of measurements to predict is high, the prediction error increase, this behavior can be observed in Fig. 5, where error estimation curve is modeled.

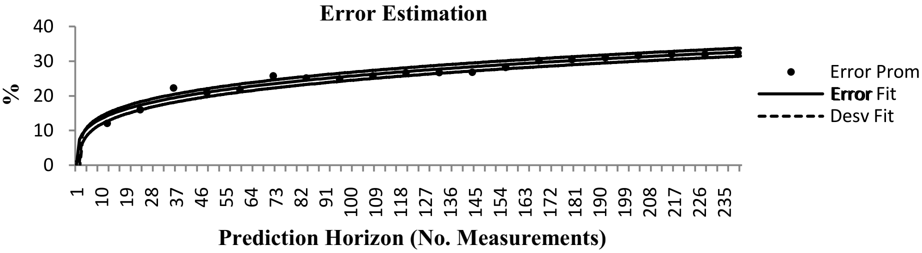


Fig. 5. Error estimation for different prediction horizon

The error prediction can be estimated with the next equation:

$$e = 7.40362x^{0.270686} \pm 1.16313 \tag{18}$$

where *e* is the predicted error, *x* is the prediction horizon expressed as the number of predicted points, and the term 1.16313 is the average deviation of the prediction error.

6 Conclusions

In this work a model for predicting the status of the ecological environment for aquaculture systems was developed. The model to predict the water quality status was built in two phases: the first predicts a section of the four signals; the second analyzes the four sections with the HWQI index in order to analyze the predicted signals and to create the PWQI index. A comparison between model (HWQI and PWQI) shows a good performance of the prediction process and the error analysis shows how a bigger horizon prediction increases the error with a light deviation. The model proposed in this research is a powerful tool in the decision support for monitoring future environmental problems in aquaculture shrimp systems.

References

1. Martínez Córdova Rafael, M.: Cultivo de Camarones Pendidos. In: Principios y Practicas, AGT Editor S.A. (1994)
2. Hirono, Y.: Current practices of water quality management in shrimp farming and their limitations. In: Proceedings of the Special Session on Shrimp Farming. World Aquaculture Society, USA (1992)
3. [ACA] Agencia Catalana del Agua, Catalonia, Spain (2005), http://www.mediambient.gencat.net/aca/ca/inici.jsp (accessed August 2007)
4. [NSF] National Sanitation Foundation International (2005), http://www.nsf.org (accessed August 2007)
5. [CCME] Canadian Council of Ministers of the Environment (Canada). An assessment of the application and testing of the water quality index of the Canadian Council of Ministers of the Environment for selected water bodies in Atlantic Canada. National indicators and reporting office (2004), http://www.ec.gc.ca/soer-ree/N (accessed August 2007)
6. Kenneth, H.: Water Quality Prediction and Probability Network Models. North Carolina State University (1998)
7. Emmanuel, C.: Digital signal processing: a practical approach. Addison-Wesley, Reading (1993)
8. Chapra, S., Canale, R.: Métodos Numéricos para Ingenieros. McGraw-Hill, México (1999)
9. Cohen, L.: Time-frequency signal analysis. Prentice Hall PTR, Englewood Cliffs (1995)

VII Document Processing and Recognition

Learning Co-relations of Plausible Verb Arguments with a WSM and a Distributional Thesaurus*

Hiram Calvo[1,2], Kentaro Inui[2], and Yuji Matsumoto[2]

[1] Center for Computing Research, National Polytechnic Institute, DF, 07738, Mexico
[2] Nara Institute of Science and Technology, Takayama, Ikoma, Nara 630-0192, Japan
{calvo,inui,matsu}@is.naist.jp

Abstract. We propose a model based on the Word Space Model for calculating the plausibility of candidate arguments given one verb and one argument. The resulting information can be used in co-reference resolution, zero-pronoun resolution or syntactic ambiguity tasks. Previous work such as Selectional Preferences or Semantic Frames acquisition focuses on this task using supervised resources, or predicting arguments independently from each other. On this work we explore the extraction of plausible arguments considering their co-relation, and using no more information than that provided by the dependency parser. This creates a data sparseness problem alleviated by using a distributional thesaurus built from the same data for smoothing. We compare our model with the traditional PLSI method.

1 Introduction

Several tasks such as co-reference resolution, zero-pronoun resolution or syntactic ambiguity can be regarded as sentence reconstruction tasks that can be solved by measuring the plausibility of each candidate argument. This kind of tasks relies on resources such as semantic frames or selectional preferences for finding the most plausible candidate for a missing part given a context. Consider for example the following sentence:
There is hay at the farm. The cow eats it
We would like to connect *it* with *hay*, and not with *farm*. From selectional preferences we know that the object of *eat* should be something *edible*, so that we can say that *hay* is more *edible* than *farm*, solving this issue. From semantic frames, we have similar knowledge, but in a broader sense—there is an *ingestor* and an *ingestible*.

However, this information can be insufficient in some cases where the selectional preference depends on other arguments from the clause. For example:
The cow eats hay but the man will eat it
In this case, it is not enough information to know that *it* should be *edible*, but also the resolution depends on *who* is eating. In this case it's unlikely that *the man* eats *hay*, so the sentence might refer to the fact that he will eat *the cow*. The same happens with

* We thank the support of Mexican Government (SNI, SIP-IPN, COFAA-IPN, and PIFI-IPN), CONACYT; and the Japanese Government. The second author is currently a JSPS fellow.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 363–370, 2009.

other of arguments for verbs. For example, some of the FrameNet peripheral arguments for the ingestion frame are *instrument*, and *place*. However, there are some things which are ingested with some instrument —e.g. soup is eaten with a *spoon*, while rice is eaten with*fork*, or *chopsticks*, depending on who is eating; or at different places. Plausible argument extraction allows constructing a database dictionary of this kind of information, which can be regarded as common sense from the fact that it is possible to learn what kind of activities are performed by groups of entities automatically from large blocks of text. See Fig. 1.

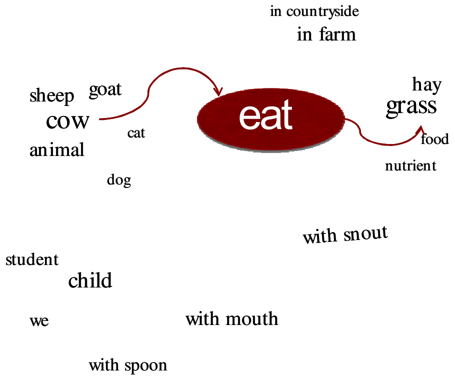


Fig. 1. A verb linking groups of related arguments

The goal of our work is to construct such a database. For this purpose we need to obtain information related to selectional preferences and semantic frames extraction.

Several works are devoted to semantic plausibility extraction, but most of them use supervised resources or consider arguments independently from each other. This work is devoted to the extraction of co-related plausible arguments in an unsupervised way, *i.e.*, no other resource is needed after the dependency parse.

The following section describes work related to verb argument plausibility acquisition, and then we present the results of our experiments within two different approaches. On Section 2.5 we briefly discuss some possible applications, on Section 3 we evaluate our approach and finally we draw or conclusions.

2 Related Work

The problem of automatic verb argument plausibility acquisition can be studied from several points of view. From the viewpoint of the kind of information extracted we can find related work for selectional preferences and semantic frames extraction. From the approach of selectional preferences, the task is focused on automatically obtaining classes of arguments for a given verb and a syntactic construction. From the approach of semantic frames, arguments are grouped by the semantic role they have, regardless of the syntactic construction they have. This latter approach emphasizes the distinction between core (indispensable) or peripheral arguments. On the other hand, we can consider the viewpoint of how this information is represented: the task can be regarded as a case of statistic language modeling, where given a context—verb and other arguments, the missing argument should be inferred with high probability; or it can be regarded as a word space model task frequently seen in IR systems. In the next sections we present works related to this task from those different viewpoints.

2.1 Selectional Preferences

Selectional preferences acquisition can be regarded as one of the first attempts to automatically find argument plausibility. Early attempts dealt with simpler <*verb*, *argument*> pairs. Since the learning resource is sparse, all of these works use a generalization, or *smoothing* mechanism for extending coverage. [16] uses WordNet for generalizing the object-argument. [1] use a class-to-class model, so that the both verb as well as the object-argument are generalized by belonging to a class using WordNet. [11] acquire selectional preferences as probability distributions over the WordNet noun hyponym hierarchy. They use other argument relationships aside from object-argument. [13] combine semantic and syntactic information by estimating his model using corpora with semantic role annotation (*i.e.* FrameNet, PropBank), and then applying class-based smoothing using WordNet. They model the plausibility of a verb and argument in a given role as $Plausibility_{v,r,a}=P(v,s,gf,r,a)=P(v)\cdot P(s|v)\cdot P(gf|v,s)\cdot P(r|v,s,gf)\cdot P(a|v,s,gf,r)$, where $P(s|v)$is the probability of a particular sense of a verb, $P(gf|v,s)$ is the syntactic subcategorizations of a particular verb sense, $P(r|v,s,gf)$ reflects how the verb prefers to realize its thematic role fillers syntactically and $P(a|v,s,gf,r)$ is verb's preference for certain argument types and estimate the fit of a verb and argument in a given role.

2.2 Subcategorization Frames

The following works deal with the problem of semisupervised argument plausibility extraction from the subcategorization frames extraction approach. [17] acquire verb argument structures. They generalize nouns by using a Named Entity Recognizer (IdentiFinder) and then they use the noisy channel framework for argument prediction. Example of the kind of information they are working with are: ***Organization*** *bought* ***organization*** *from* ***organization*****,** ***Thing*** *bought the outstanding shares on* ***date*****,** and, sometimes without generalization**,** *The cafeteria bought extra plates*.

Another semi-supervised work is [9]. They generalize by using a manually created thesaurus. For finding case frames they use together with the verb, the closest argument, providing verb sense disambiguation for cases similar as the example which motivated us, presented in Section 1.

Next we discuss two different viewpoints for dealing with the verb argument information representation.

2.3 Language Modeling

We can regard the task of finding the plausibility of a certain argument for a set of sentences as estimating a word given an specific context. Particularly for this work we can consider context as the grammar relationships for a particular verb:

$$P(w,c) = P(c) \cdot P(c|w) \tag{1}$$

which can be estimated in many ways, particularly, using a hidden markov model, or using latent variables for smoothing, for PLSI [8]:,

$$P(w,c) = \sum_{Z_i} P(z) \cdot P(w|z) \cdot P(c|z)$$

The conditional probability can be calculated from n-gram frequency counts.

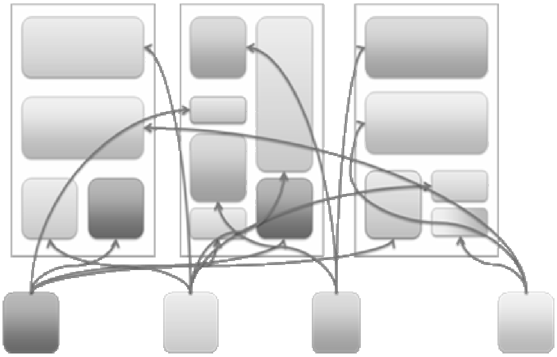


Fig. 2. Each document in PLSI is represented as a mixture of topics

2.4 Word Space Model

Traditionally from Information Retrieval, words can be represented as documents and semantic context as features, so that it is possible to build a co-occurrence matrix, or word space, where each intersection of word and context shows the frequency count of each number. This approach has been recently used with syntactic relationships [13]. An important issue within this approach is the similarity measure chosen for comparing words (documents) given its features. Popular similarity measures range from simple measures such as Euclidean distance, cosine and Jaccard's coefficient [10], to measures such as Hindle's measure and Lin's measure.

2.5 Potential Applications

Since Resnik [16], selectional preferences have been used in a wide range of applications: Improving parsing, since it is possible to disambiguate syntactic structures if we know the kind of arguments expected for a sentence [4]; Inference of meaning of unknown words—Uttering I eat *borogoves* makes us think that a *borogove* might be edible; Co-reference resolution [15]; Word Sense Disambiguation [11, 12]; Metaphora recognition, an uncommon usage of an argument would make a sentence odd, thus, perhaps containing a metaphora (or a coherence mistake); Semantic Plausibility [14], Malapropism detection [2, 3] *"hysteric center"*, instead of *historic center, "density* has brought me to you", instead of *destiny*.

3 Methodology

We propose a model based on the Word Space Model. Since we want to consider argument co-relation, we use the following information:

$P(v,r_1,n_1,r_2,n_2)$, where v is a verb, r_1 is the relationship between verb and n_1 (noun) as subject, object, preposition or adverb. r_2 and n_2 are analogous. If we assume that n has a different function when used with another relationship, then we can consider that r and n form a new symbol, called a. So that we can simplify our 5-tuple to $P(v,a_1,a_2)$. We want to know, given a verb and an argument a_1, which a_2 is the most plausible, we can write this as $P(a_2|v,a_1)$. For PLSI this can be estimated by

$P(a_2,v,a_1)$=Sum(Z_i,$P(z)\cdot P(a_2|z)\cdot P(v,a_1|z)$).

For the word space model, we can build a matrix where a_2 are the rows (documents) and v, a_1 are features. As this matrix is very sparse, we use a thesaurus for smoothing the argument values. For doing this, we loosely followed the approach proposed by [12] for finding the predominant sense, but in this case we use the k nearest neighbors of each argument a_i to find the prevalence score of an unseen triple given its similarity to all triples present in the corpus, measuring this similarity between arguments. In other words, as in [12, 18, 19] for WSD, each similar argument votes for the plausibility of each triple.

$$Prevalence(v,x_1,x_2) = \frac{\sum\limits_{<v,a_1,a_2>\in T} sim(a_1,x_1)\cdot sim(a_2,x_2)\cdot P_{MLE}(v,a_1,a_2)}{\sum\limits_{<v,a_1,a_2>\in T} sim\_exists(a_1,a_2,x_1,x_2)}$$

where T is the whole set of < *verb*, *argument*$_1$, *argument*$_2$> triples and

$$sim\_exists(a_1,a_2,x_1,x_2) = \begin{cases} 1 & \text{if } sim(a_1,x_1)\cdot sim(a_2,x_2) > 0 \\ 0 & \text{otherwise} \end{cases}$$

For measuring the similarity between arguments, we built a thesaurus using the method described by [5], using the Minipar browser [6] over short-distance relationships, *i.e.*, we previously separated subordinate clauses. We obtained triples $<v,a_1,a_2>$ from this corpus, which were counted, and these were used for both building the thesaurus as well as a source of verb and argument co-occurrences.

3.1 Evaluation

We compared these two models in a pseudo-disambiguation task following [20]. First we obtained triples $\langle v,a_1,a_2\rangle$ from the corpus. Then, we divided the corpus in training (80%) and testing (20%) parts. With the first part we trained the PLSI model and created the WSM. This WSM was also used for obtaining the similarity measure for every pair of arguments a_2, a_2' . Then we are able to calculate $Plausibility(v, a_1, a_2)$. For evaluation we created artificially 4-tuples: $\langle v, a_1, a_2, a_2' \rangle$, formed by taking all the triples $\langle v, a_1, a_2 \rangle$ from the testing corpus, and generating an artificial tuple $\langle v, a_1, a_2' \rangle$ choosing a random a_2' with $r_2' = r_2$, and making sure that this new random triple $\langle v, a_1, a_2' \rangle$ was not present in the training corpus. The task consisted on selecting the correct tuple.

We compared two models based on the Statistical Language Model and the Word Space Model approaches respectively. Using the patent corpus from the NII Test Collection for Information Retrieval System, NTCIR-5 Patent [7], we parsed 7300 million tokens, and then we extracted the chain of relationships on a directed way, that is, for the sentence: X add Y to Z by W, we extracted the triples: <*add, subj-X, obj-Y*>, <*add, obj-Y, to-Z*>, <*add, to-Z, by-W*>. We obtained 706M triples in the form <v, a_1, a_2>. We considered only chained asymmetric relationships to avoid false similarities between words co-occurring in the same sentence.

Following [20], we chose 20 verbs, covering high-frequency verbs and low-frequency verbs and for each one we extracted all the triples <v, a_1, a_2> present in the triples corpus. Then we performed experiments with the PLSI algorithm, and the WSM algorithm.

We experimented with different number of topics for the latent variable z in PLSI, and with different number of neighbors from the Lin thesaurus for expanding the WSM. Results are shown in Table 1 for individual words, 10 neighbors for WSM and 10 topics for PLSI. Figure 3 shows average results for different neighbors and topics.

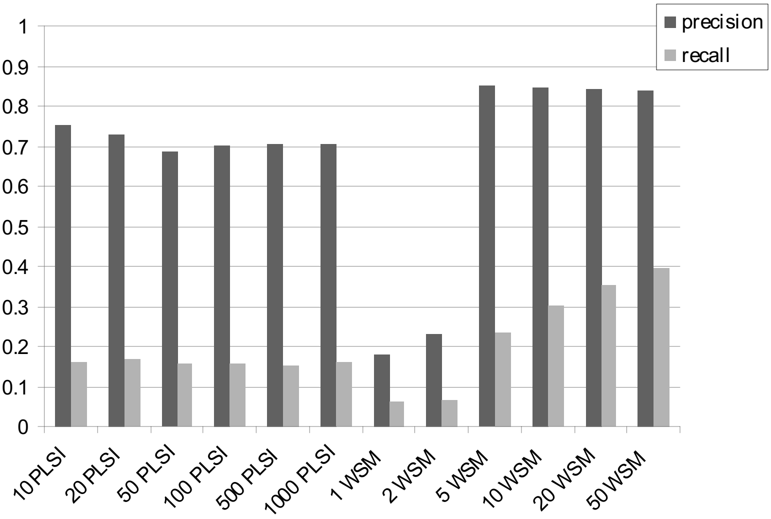


Fig. 3. Results for (topics)-PLSI and (neighbors)-WSM

4 Conclusions

We have proposed a new algorithm within the WSM approach for unsupervised plausible argument extraction and compared with a traditional PLSI approach, obtaining particular evidence to support that it is possible to achieve better results with the method which votes for common triples using a distributional thesaurus. The results look consistent with previous works using distributional thesauri [4,18,19] (see Figure 3): adding information increases coverage with little sacrifice on precision. We should experiment with the upper limit of the increasing coverage, as each neighbor from the thesaurus is adding noise. We have experimented with building the thesaurus using the same corpus; however, significant differences could be found if using an encyclopedia corpus for building the dictionary, as broader and richer context could be found.

We call our work unsupervised because we are not using any other resource after the dependency parser, such as named entity recognizers, or labeled data used for training a machine learning algorithm. As a future work we plan to improve the overall recall measure by adding a back-off technique, which could consist simply on considering the information based on the verb when information of the verb and one argument is not available. We plan also to explore specific applications for this algorithm.

Table 1. Precision (P) and Recall (R) for each verb for 10 neighbors (WSM) and 10 topics (PLSI)

verb	**triples**	**WSM-10**		**PLSI-10**	
		P	**R**	**P**	**R**
eat	31	0.98	0.92	1.00	0.04
seem	77	0.88	0.09	0.64	0.38
learn	204	0.82	0.10	0.57	0.22
inspect	317	0.84	0.19	0.43	0.12
like	477	0.79	0.13	0.54	0.24
come	1,548	0.69	0.23	0.78	0.17
play	1,634	0.68	0.18	0.69	0.19
go	1,901	0.81	0.25	0.80	0.15
do	2,766	0.80	0.24	0.77	0.19
calculate	4,676	0.91	0.36	0.81	0.13
fix	4,772	0.90	0.41	0.80	0.13
see	4,857	0.76	0.23	0.84	0.20
write	6,574	0.89	0.31	0.82	0.15
read	8,962	0.91	0.36	0.82	0.11
add	15,636	0.94	0.36	0.81	0.10
have	127,989	0.95	0.48	0.89	0.03
average	11,401	0.85	0.30	0.75	0.16

References

1. Agirre, E., Martinez, D.: Learning class-to-class selectional preferences. In: Workshop on Computational Natural Language Learning, ACL (2001)
2. Bolshakov, I.A., Galicia-Haro, S.N., Gelbukh, A.F.: Detection and Correction of Malapropisms in Spanish by Means of Internet Search. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 115–122. Springer, Heidelberg (2005)
3. Budanitsky, E., Graeme, H.: Semantic distance in WorldNet: An experimental, application-oriented evaluation of five measures. In: NAACL Workshop on WordNet and other lexical resources (2001)
4. Calvo, H., Gelbukh, A., Kilgarriff, A.: Automatic Thesaurus vs. WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 177–188. Springer, Heidelberg (2005)
5. Lin, D.: Automatic Retrieval and Clustering of Similar Words. In: Procs. 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics (1998)
6. Lin, D.: Dependency-based Evaluation of MINIPAR. In: Proc. Workshop on the Evaluation of Parsing Systems (1998)
7. Fujii, A., Iwayama, M. (eds.): Patent Retrieval Task (PATENT). Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (2005)
8. Hoffmann, T.: Probabilistic Latent Semantic Analysis, Uncertainity in Artificial Intelligence, UAI (1999)
9. Kawahara, D., Kurohashi, S.: Japanese Case Frame Construction by Coupling the Verb and its Closest Case Component. In: 1st Intl. Conf. on Human Language Technology Research, ACL (2001)
10. Lee, L.: Measures of Distributional Similarity. In: Procs. 37th ACL (1999)
11. McCarthy, D., Carroll, J.: Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. Computational Linguistics 29(4), 639–654 (2006)
12. McCarthy, D., Koeling, R., Weeds, J., Carroll, J.: Finding predominant senses in untagged text. In: Procs. 42nd meeting of the ACL, pp. 280–287 (2004)
13. Padó, S., Lapata, M.: Dependency-Based Construction of Semantic Space Models. Computational Linguistics 33(2), 161–199 (2007)
14. Padó, U.M., Crocker, F., Keller, F.: Modelling Semantic Role Plausibility in Human Sentence Processing. In: Procs. EACL (2006)
15. Ponzetto, P.S., Strube, M.: Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In: Procs. Human Language Technology Conference, NACACL, pp. 192–199 (2006)
16. Resnik, P.: Selectional Constraints: An Information-Theoretic Model and its Computational Realization. Cognition 61, 127–159 (1996)
17. Salgueiro, P., Alexandre, T., Marcu, D., Volpe Nunes, M.: Unsupervised Learning of Verb Argument Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 59–70. Springer, Heidelberg (2006)
18. Tejada, J., Gelbukh, A., Calvo, H.: An Innovative Two-Stage WSD Unsupervised Method. SEPLN Journal 40 (March 2008)
19. Tejada, J., Gelbukh, A., Calvo, H.: Unsupervised WSD with a Dynamic Thesaurus. In: 11th International Conference on Text, Speech and Dialogue. TSD 2008, Brno, Czech Republic, September 8–12 (2008)
20. Weeds, J., Weir, D.: A General Framework for Distributional Similarity. In: Procs. conf. on EMNLP, vol. 10, pp. 81–88 (2003)

Handwritten Word Recognition Using Multi-view Analysis

J.J. de Oliveira Jr.[1], C.O. de A. Freitas[2],
J.M. de Carvalho[3], and R. Sabourin[4]

[1] UFRN - Universidade Federal do Rio Grande do Norte
josejosemarjr@gmail.com
[2] PUC-PR - Pontíficia Universidade Católica do Paraná
cinthia.freitas@pucpr.br
[3] UFCG - Universidade Federal de Campina Grande
carvalho@dee.ufcg.edu.br
[4] ÉTS - École de Technologie Supérieure
sabourin@etsmtl.ca

Abstract. This paper brings a contribution to the problem of efficiently recognizing handwritten words from a limited size lexicon. For that, a multiple classifier system has been developed that analyzes the words from three different approximation levels, in order to get a computational approach inspired on the human reading process. For each approximation level a three-module architecture composed of a zoning mechanism (pseudo-segmenter), a feature extractor and a classifier is defined. The proposed application is the recognition of the Portuguese handwritten names of the months, for which a best recognition rate of 97.7% was obtained, using classifier combination.

1 Introduction

In a general way handwritten recognition systems are defined by two operations: features extraction and classification. Feature extraction is related to information extraction, creating the word representation used as input to the classifier. Thus, the goal of feature extraction is to capture the most relevant and discriminatory information of the object to be recognized, eliminating redundancies and reducing the data amount to be processed. The classifier based on this representation associates conditional probabilities to the classes by means of an estimation process.

This study deals with recognition of the Portuguese month names represented by a limited lexicon of 12 classes: Janeiro, Fevereiro, Março, Abril, Maio, Junho, Julho, Agosto, Setembro, Outubro, Novembro and Dezembro. Some of these classes share a common sub-string, which adds complexity to the problem. As can be observed in Figure 1, there is similarity between the suffix of some classes in the lexicon, which creates confusion and affects the performance of the recognizer. Another source of confusion is a common first letter (e.g. junho and

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 371–378, 2009.

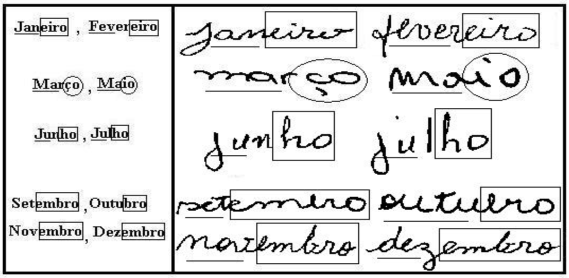


Fig. 1. Complexity of the recognition problem: prefix and suffix

julho), which plays a significant role in the word recognition process, as observed by Schomaker[1]. Further difficulty is added by the fact that the vowels (a, e, i, o) exhibit low discriminatory power in the human reading process[1].

Performance of the recognition system for the considered lexicon is limited by the type of confusion illustrated in Figure 1. To overcome these constraints, we utilize an approach based on multi-view representation and perceptual concepts, designed to avoid the intrinsic difficulties of the lexicon. This approach is one evolution of other previous systems published by the same research group[2,3], however the multi-view analysis proposed here are original and it is based on perceptual concepts. This particular choice of lexicon, does not take from the generality of the solution, since that the same problem are founded in other lexicons like in French language[4]. The proposed system can be applied equally well to any similar problem, thus bringing a true contribution to the state of the art in the area.

This paper is divided into 4 sections. Section 2 describes the overall system developed, considering the multi-view representation and system architecture. In Section 3, the experimental results are presented and analyzed. Finally in Section 4 the conclusions and suggestions for future work are presented.

2 Methodology

This section presents an overview of the proposed system based on multi-view representation. The words database used and the preprocessing operations applied are described. Next, each pseudo-segmentation scheme is defined, combined with feature vectors extraction and the classification method utilized. The classifiers outputs are combined in order to produce a final decision for the sample in analysis.

2.1 Multi-view Analysis

Usually, two main approaches are considered for Handwritten Word Recognition - HWR problems: local or analytical approaches held at character level and global approach held at word level[5].

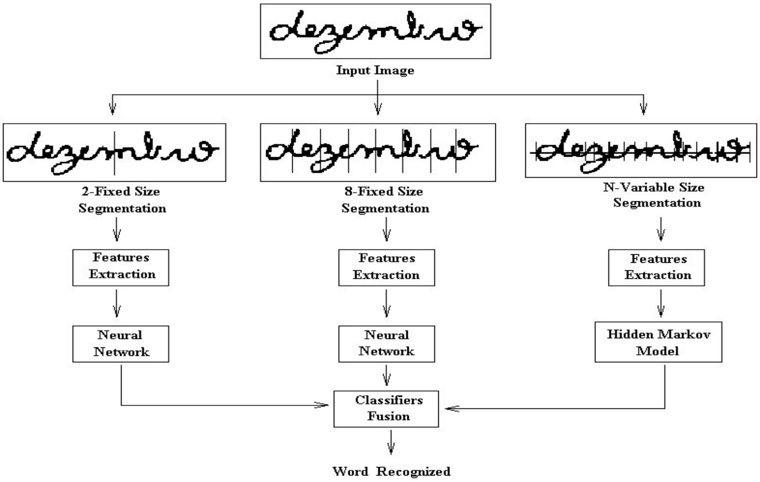


Fig. 2. System block diagram

The global approach extracts features from the words as a whole, therefore making it unnecessary to explicitly segment words into characters or pseudo-characters. This approach seeks to explore information from the word context, allowing aspects based on psychological models to be considered. The global or word level allows to incorporate principles of the human reading process into computational methods[5].

The local approach utilizes the word basic elements in the recognition strategy. These elements can be characters or segments of characters (pseudo-characters). This approach is characterized by the difficulty to define the segmentation or separation points between characters. Therefore, success of the recognition method will depend on success of the segmentation process[5].

Our focus is on the global level, as it seeks to understand the word as a whole, similarly to the human reading process where the reader uses previous knowledge about the word shape to perform recognition.

Although the global analysis does not use segmentation, pseudo-segmentation (or zoning) mechanisms can be added to produce a robust recognition system[3]. Zoning basically consists of partitioning the word image into segments (or sub-images) of equal or variable size. Three different zoning schemes have been employed, as described next and illustrated in Figure 2.

- 2-FS (2 fixed size sub-regions): Each image is split in two, to the right and to the left of the word center of gravity[2];
- 8-FS (8 fixed size sub-regions): Each image is divided in 8 sub-regions of equal size. This number corresponds to the average number of letters in the lexicon;
- N-VS (N variable size sub-regions): The words horizontal projection histogram of black-white transitions is determined. The line with maximum

histogram value is called Central Line (CL). A segment is defined by two consecutive transitions over the CL.

Multi-view analysis therefore, seeks to provide different simultaneous approximations for the same image. For each zoning procedure, one specific feature vector and classifier are defined, all based on global word interpretation. At the end, the classifiers outputs are combined to produce the final decision, therefore taking advantage of the zoning mechanisms complementarity.

2.2 Word Database and Preprocessing

To develop the system it was initially necessary to construct a database that can represent the different handwriting styles present in the Brazilian Portuguese language for the chosen lexicon. The words were digitized at 200 dpi. Figure 3 illustrates some samples from this database. To reduce the variability, slant and baseline skew normalization algorithms were applied, using inclinated projection profiles and shear transformation.

2.3 2-FS Feature Set

In this word representation, perceptual features and characteristics based on contour as concavities/convexities are represented by the number of their occurrences. The features extracted from each word form a vector of dimension 24. Perceptual features are considered high-level features due to the important role they play in the human reading process, which uses features like ascenders, descenders and estimation of word length to read handwritten words[5].

The components of the feature set can be described as following[2]:

- Number of concave and convex semicircles, number of horizontal and vertical lines, number of ascenders and descenders with loop in the left/right areas, respectively;
- Number of crossing-points, branch-points, end-points, loops, ascenders and descenders on the left/right areas, respectively;
- Number of horizontal axis crossings by stroke;
- Proportion of white/black pixels inside the word bounding box.

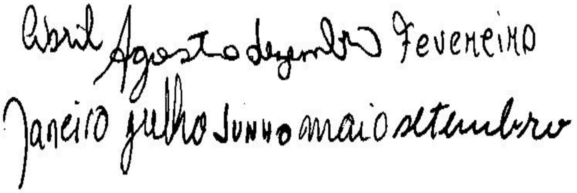


Fig. 3. Sample images from the database

2.4 8-FS Perceptual Feature Set (P)

In this zoning mechanism, ten patterns are defined for each sub-region, thus forming for each image a feature vector containing 80 patterns.

The 10 patterns used in the perceptual feature set are:

- x_1, x_2, x_3, x_4 **- Ascender (and Descender) position and size:** Position and height of the ascender (and descender) central pixel;
- x_5, x_6, x_7 **- Closed loop size and location:** Number of pixels inside a closed loop and coordinates of the closed loop center of mass;
- x_8, x_9 **- Concavity measure:** Initially the convex hull is constructed starting at the bottom-most point of the boundary. The leftmost and rightmost points in the hull are detected and the angles (relative to the horizontal) defined by the line segments joining them to the starting point are measured;
- x_{10} **- Estimated segment length:** Number of transitions (black-white) in the central line of the sub-region outside of the closed loops.

2.5 8-FS Directional Feature Set (D)

The directional features can be considered intermediate-level features, conveying relevant information about the image background[5]. In this paper, the directional features defined are based on concavity testing, where for each white image pixel (or background pixel) it is tested which of the four main directions (NSEW) leads to a black (contour) pixel.

Representation is made labeling the background pixels, that depends on the combination of the open directions. The components of the feature vector for each sub-region are obtained by counting the number of pixels for each label.

2.6 2-FS and 8-FS Classifier

To each 2-FS and 8-FS features set one classifier based on Class-Modular MLP was defined. It follows the principle that a single task is decomposed into multiple subtasks and each subtask is allocated to an expert network. In this paper, as well as in Oh et al.[6], the K-classification problem is decomposed into K 2-classification subproblems. For each one of the K classes, one 2-classifier is specifically designed.

Therefore, the 2-classifier discriminates that class from the other $K-1$ classes. In the class-modular framework, K 2-classifiers solve the original K-classification problem cooperatively and the class decision module integrates the outputs from the K 2-classifiers.

2.7 N-VS Feature Set

The features utilized by the N-VS zoning mechanism are the same as those presented in sections 2.4 and 2.5, though different extraction and representation methods were used and adapted to this approach. A symbol is designated to represent the extracted set of features for each segment, building up a grapheme.

In the case where no feature is extracted from the analyzed segment, an empty symbol denoted by **X** is emitted. This feature set is capable of representing the link between letters and separating graphemes[4,7].

2.8 N-VS Classifier

The N-VS zoning mechanism defines a variable number of sub-regions which makes neural network application difficult. Therefore, Hidden Markov Model (HMM) classifiers are more recommended in this case[4,7]. The entire and definitive alphabet is composed of 29 different symbols selected from all possible symbol combinations, using the mutual information criterion[4,7].

Our HMM word models are based on a left-right discrete topology where each transition can skip at the most two states. Model training is based on the Baum-Welch Algorithm and the Cross-Validation process is performed on two data sets: training and validation. After the Baum-Welch Algorithm iteration on the training data, the likelihood of the validation data is computed using the Forward Algorithm[4,7]. During the experiments, the matching scores between each model λ_i and an unknown observation sequence O are carried out using the Forward Algorithm.

3 Experimental Results

For the experiments, the database was randomly split into three data sets: Set 1 - Training Base with 6,120 words; Set 2 - Validation Base and Set 3 - Testing Base, both with 2,040 words. For each set, the words are evenly distributed among the classes.

For each feature set considered in the system (2-FS and 8-FS), 12 (twelve) Class-Modular MLP classifiers were trained and tested. In the Class-Modular approach, the classifier that presents the maximum output value indicates the class recognized[6]. The amount of neurons in the hidden layer was empirically determined, different configurations being tested. Each K 2-classifier is independently trained using the training and validation sets. The Back-propagation Algorithm was used in all cases. To train a 2-classifier for each word class, we reorganized the original training and validation sets into 2 sub-sets: Z_0 that contains the samples from current class and Z_1 that contains the samples from all other K-1 classes. To recognize the input patterns, the class decision module considers only the O_0 outputs from each sub-network and uses a simple winner-takes-all scheme to determine the recognized class[6].

The N-VS classifier was evaluated with the N-VS feature set and for each class one HMM was trained and validated. The model that assigns maximum probability to one test image represents the class recognized.

Table 1 shows the results obtained for each classifier individually. It can be seen that the best result was obtained using 8-FS classifier with directional features.

Table 1. Recognition rate obtained for each classifier individually

Classifier	2-FS	8-FS(P)	8-FS(D)	N-VS
RR	73.9%	86.3%	**91.4%**	81.7%

Table 2. Recognition rate obtained using classifiers combination

Classifiers	RR (%)
2-FS and 8-FS(P)	90.5
2-FS and 8-FS(D)	94.4
8-FS(P) and 8-FS(D)	93.6
8-FS(P) and N-VS	93.5
8-FS(D) and N-VS	95.6
2-FS and N-VS	90.5
2-FS, 8-FS(P) and 8-FS(D)	95.4
2-FS, 8-FS(P) and N-VS	95.8
2-FS, 8-FS(D) and N-VS	97.2
8-FS(P), 8-FS(D) and N-VS	96.9
2-FS, 8-FS(P), 8-FS(D) and N-VS	97.7

3.1 Classifiers Fusion

To obtain the hybrid classifier it is necessary to define a combination rule for the classifiers's outputs. Initially, we made the assumption that an object Z must be assigned to one of the K possible classes $(w_1, \cdots, w_K)$ and assume that L classifiers are available, each one representing the given pattern by a distinct measurement vector. Denote the measurement vector used by the ith classifier as x_i and the *a posteriori* probability $P(w_j|x_1, \cdots, x_L)$[8]. Therefore, the combining rule is defined as:

– Weighted Sum (WS): Assigns Z to class w_j if

$$\sum_{i=1}^{L} \alpha_i \cdot p(w_j|x_i) = \max_{k=1}^{K} \sum_{i=1}^{L} \alpha_i \cdot p(w_k|x_i); \quad (1)$$

where α_i, $i = 1, \cdots, L$ are weights for each classifier.

To guarantee that the classifier outputs represent probabilities, an output normalization was performed: $P^*(w_j|x_i) = \frac{P(w_j|x_i)}{\sum_K P(w_j|x_i)}$. The best weights were obtained by an exhaustive search procedure, considering for each classifiers combination, 2,000 different n-upla of weight vectors with random adaptation.

The average recognition rates obtained considering different classifiers combination are presented in Table 2. It can be seen that the best result was obtained using combination for 2-FS, 8-FS(P), 8-FS(D) and N-VS classifiers.

4 Discussion and Conclusions

This paper presents a hybrid system using a methodology based on multi-view analysis, applied to the recognition of the Portuguese handwritten names of the months. This system is based on a Global Approach, which extracts global features from the word image, avoiding explicit segmentation. This approach is more that one simple combination of classifiers since that explores word context information, while allows incorporating aspects based on perceptual concepts. Therefore, unlike other proposed systems, we have a computational approximation inspired in the human reading process.

We have evaluated the efficiency of multiple architectures using Neural Network and Hidden Markov Models classifiers for the handwritten word recognition problem. The main conclusion obtained is that the analyzed classifiers are complementary and the combining strategy proposed enhances their complementarity. Therefore, the classifiers arranged in the multi-view analysis are a better solution for our problem than any of the classifiers applied individually. This result indicates that a similar strategy can be applied to other restricted lexicons. Future work will focus on the analysis of adaptative models that will be applied to large lexicons.

References

1. Schomaker, L., Segers, E.: A Method for the Determination of Features used in Human Reading of Cursive Handwriting. In: IWFHR 1998, The Netherlands, pp. 157–168 (1998)
2. Kapp, M.N., de Almendra Freitas, C.O., Sabourin, R.: Methodology for the Design of NN-based Month-Word Recognizers Written on Brazilian Bank Checks. International Journal on Image and Vision Computing 25(1), 40–49 (2007)
3. de Almendra Freitas, C.O., Oliveira, L.S., Aires, S.K., Bortolozzi, F.: Handwritten Character Recognition Using Non-Symmetrical Perceptual Zoning. International Journal on Pattern Recognition and Artificial Intelligence 21(1), 1–21 (2007)
4. de Almendra Freitas, C.O., Bortolozzi, F., Sabourin, R.: Study of Perceptual Similarity Between Different Lexicons. International Journal on Pattern Recognition and Artificial Intelligence 18(7), 1321–1338 (2004)
5. Madhvanath, S., Govindaraju, V.: The Role of Holistic Paradigms in Handwritten Word Recognition. IEEE Trans. on PAMI 23(2), 149–164 (2001)
6. Oh, I., Suen, C.-Y.: A Class-Modular Feedforward Neural Network for Handwriting Recognition. Pattern Recognition 35(1), 229–244 (2002)
7. de Almendra Freitas, C.O., Bortolozzi, F., Sabourin, R.: Handwritten Isolated Word Recognition: An Approach Based on Mutual Information for Feature Set Validation. In: ICDAR 2001, Seattle - USA, pp. 665–669 (2001)
8. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. IEEE Trans. on PAMI 20(3), 226–239 (1998)

A Speed-Up Hierarchical Compact Clustering Algorithm for Dynamic Document Collections

Reynaldo Gil-García and Aurora Pons-Porrata

Center for Pattern Recognition and Data Mining
Universidad de Oriente, Santiago de Cuba, Cuba
{gil,aurora}@cerpamid.co.cu

Abstract. In this paper, a speed-up version of the *Dynamic Hierarchical Compact* (*DHC*) algorithm is presented. Our approach profits from the cluster hierarchy already built to reduce the number of calculated similarities. The experimental results on several benchmark text collections show that the proposed method is significantly faster than *DHC* while achieving approximately the same clustering quality.

Keywords: hierarchical clustering, dynamic clustering.

1 Introduction

The World Wide Web and the number of text documents managed in organizational intranets continue to grow at an amazing speed. Managing, accessing, searching and browsing large repositories of text documents require efficient organization of the information. In dynamic information environments, such as the World Wide Web or the stream of newspaper articles, it is usually desirable to apply adaptive methods for document organization such as clustering.

Dynamic algorithms have the ability to update the clustering when data are added or removed from the collection. These algorithms allow us dynamically tracking the ever-changing large scale information being put or removed from the web everyday, without having to perform complete re-clustering.

Hierarchical clustering algorithms have an additional interest, because they provide data-views at different levels of abstraction, making them ideal for people to visualize and interactively explore large document collections. Besides, clusters very often include sub-clusters, and the hierarchical structure is indeed a natural constraint on the underlying application domain. In the context of hierarchical document clustering the high dimensionality of the data and the large size of text collections are two of the major challenges facing researchers today.

In [1] a hierarchical clustering algorithm, namely *Dynamic Hierarchical Compact* (*DHC*), was proposed. This method is not only able to deal with dynamic data while achieving a similar clustering quality than static state-of-the-art hierarchical algorithms but also has a linear computational complexity with respect to the number of dimensions. It uses a multi-layered clustering to update the hierarchy when new documents arrive (or are removed). The process in each layer

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 379–386, 2009.

involves two steps: the building of similarity-based graphs and the obtaining of the connected components for these graphs. The graph construction requires to perform range similarity queries, that is, given a new document d and a similarity threshold β, to retrieve all documents whose similarity to d is greater than or equal to β. These documents are called β-similar. *DHC* needs to compute the similarities between the new document and all existing documents, which is the most time-consuming operation.

In the literature, many access methods such as *M-Tree* [2] and *IQ-Tree* [3] have been proposed to efficiently perform range similarity queries. Most of them are based on a tree structure, which is traversed to find the β-similar objects to a given one. These methods partition the data sets and use the triangle unequality property to prune the search space. Unfortunately, they have bad performance in very high-dimensional and sparse spaces. This effect has been named, by researchers in the area, the curse of dimensionality. The access methods can be classified into two categories: exact and approximate. The first find the same β-similar objects that would be found using the exhaustive search, whereas the second do not guarantee to find them but they obtain an approximation faster than the exact methods.

In this paper, we present a speed-up version of the *DHC* algorithm for clustering of dynamic document collections. Following the idea of access methods, this approach profits from the cluster hierarchy already built to reduce the number of calculated similarities. It uses an approximate strategy for computing the β-similar clusters to a given one. The experimental results on several benchmark text collections show that the proposed method is significantly faster than *DHC* while achieving approximately the same clustering quality.

The remainder of the paper is organized as follows: Section 2 describes the speed-up *DHC* clustering algorithm. The evaluation carried out over six text collections is shown in Section 3. Finally, conclusions are presented in Section 4.

2 Speed-Up Dynamic Hierarchical Compact Algorithm

DHC is an agglomerative method based on graph. It uses a multi-layered clustering to produce the hierarchy. The granularity increases with the layer of the hierarchy, with the top layer being the most general and the leaf nodes being the most specific. The process in each layer involves two steps: construction of similarity-based graphs and obtaining the connected components for these graphs. Each connected component represents a cluster.

DHC algorithm uses two graphs. The first one is the β-similarity graph, which is an undirected graph whose vertexes are the clusters and there is an edge between vertexes i and j, if the cluster j is β-similar to i. Two clusters are β-similar if their similarity is greater than or equal to β, where β is a user-defined parameter. Analogously, i is a β-isolated cluster if its similarity with all clusters is less than β. As inter-cluster similarity measure we use group-average (i.e., the average of the similarities between elements of the two clusters to be compared). In the vector space model, the cosine similarity is the most

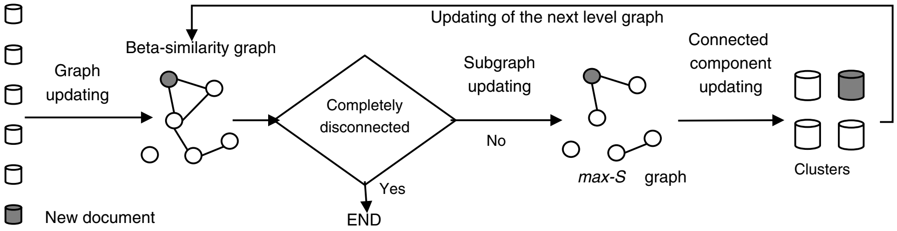


Fig. 1. Dynamic Hierarchical Compact algorithm

commonly used measure to compare the documents. By using this measure we can take advantage of a number of properties involving the composite vector of a cluster (i.e., the sum of document vectors of the cluster) [4]. In particular, the group-average similarity between clusters i and j is equal to the fraction between the scalar product of the composite vectors of these clusters and the product of clusters' sizes.

The second graph relies on the maximum β-similarity relationship (denoted as *max-S* graph) and it is a subgraph of the first one. The vertices of this graph coincide with vertices in the β-similarity graph, and there is an edge between vertices i and j, if i is the most β-similar cluster to j or vice versa.

Given a cluster hierarchy previously built by the algorithm, each time a new document arrives (or is removed), the clusters at all levels of the hierarchy must be revised (see Figure 1). When a new document arrives (or is removed), a singleton is created (or deleted) and the β-similarity graph at the bottom level is updated. Then, the *max-S* graph is updated too, which produce (or remove) a vertex and can also produce new edges and remove others. These changes on the *max-S* graph lead to the updating of the connected components. When clusters are created or removed from a level of the hierarchy, the β-similarity graph at the next level must be updated. This process is repeated until this graph is completely disconnected (all vertices are β-isolated). It is possible that the β-similarity graph became completely disconnected before the top level of the hierarchy is reached. In this case, the next levels of the hierarchy must be removed. Notice that the algorithm uses the same β value in all hierarchy levels.

The steps are shown in Algorithm 1. A detailed description of steps 4(a) and 4(b) can be seen in [1].

The updating of the β-similarity graph in *DHC* is trivial. For each vertex to add, the similarities with the remaining vertices are calculated and the corresponding edges are added to the graph. On the contrary, for each vertex to remove, all its edges are removed too. Notice that *DHC* needs to compute the similarities between the new document and all existing documents at the bottom level. Also, for each level of the hierarchy the similarities between the new clusters created at the previous level and the existing clusters at the corresponding level must be calculated too. The computation of these similarities in all levels of the hierarchy is the most time-consuming operation.

Algorithm 1. Dynamic Hierarchical Compact steps.

```
1. Arrival of a document to cluster (or to remove).
2. Put the new document in a cluster on its own (or remove the single cluster to
   which the document belongs).
3. $level = 0$ and update the $\beta$-similarity graph at the bottom level, $G_0$.
4. While $G_{level}$ is not completely disconnected:
   (a) Update the $max$-$S$ graph at $level$.
   (b) Update the connected components for the $max$-$S$ graph.
   (c) Update the $\beta$-similarity graph at the next level, $G_{level+1}$.
   (d) $level = level + 1$
5. If there exist levels greater than $level$ in the hierarchy, remove them.
```

Our proposal focuses on improving the performance of the β-similarity graph updating. The key idea is to profit from the cluster hierarchy already built to find out the possible β-similar clusters to a given one, without having to compute all similarities. Each cluster in the hierarchy is associated to its composite vector.

Given a new cluster c created at a certain level l of the hierarchy, our method traverses the hierarchy already built from the top level until the level l is reached and attempts to only explore the branches in which the β-similar clusters to c possibly appear. With the aim of discarding as many branches of the hierarchy as possible, we use a similarity threshold γ. The underlying idea is that all nodes of the hierarchy whose group-average similarity to c be lesser than γ can be discarded, since they are less likely to contain the β-similar clusters to c.

In our speed-up version, we first compute the similarity of the new cluster c with the clusters at the top level of the hierarchy. Remember that in our case, the group-average similarity between two clusters is calculated from its composite vectors. Then, those clusters that are γ-similar to c are selected and its childs are evaluated. This process is repeated for the selected childs at each level of the hierarchy. The traversal goes on until the level in which c was created is reached. Once the γ-similar clusters are found in this level, the algorithm selects those clusters that are also β-similar to c. The steps are shown in Algorithm 2.

Notice that the proposed method only computes the similarities between each new cluster at a certain level of the hierarchy and all clusters belonging to the explored branches, instead of the similarities with all existing clusters in this level. This number of clusters is much lesser, allowing to reduce the number of calculated similarities. Notice also that our method does not guarantee to find all β-similar clusters to a given one, but an approximation of them instead.

3 Experimental Results

The performance of the proposed version of Dynamic Hierarchical Compact algorithm has been evaluated using six benchmark text collections, whose general characteristics are summarized in Table 1. They are heterogeneous in terms of document size, number of topics and document distribution. Human annotators

Algorithm 2. The search of β-similar clusters to a given one.

```
Input: A cluster $c$ to be added at level $l$.
Output: The $\beta$-similar clusters of $c$.

1. Let $level$ be the index of the top level.
2. Compute the similarities between $c$ and all clusters at $level$.
3. Let $S$ be the set of clusters at $level$ that are $\gamma$-similar to $c$.
4. While $level > l$:
    (a) Let $S'$ be the set of all childs of the clusters in $S$.
    (b) Compute the similarities between $c$ and all clusters in $S'$.
    (c) Remove from $S'$ all clusters that are not $\gamma$-similar to $c$.
    (d) $S = S'$
    (e) $level = level - 1$
5. Remove from $S$ all clusters that are not $\beta$-similar to $c$.
6. Return $S$.
```

Table 1. Description of document collections

Collection	Source	Documents	Terms	Topics
hitech	San Jose Mercury[a]	2301	13170	6
eln	TREC-4[b]	5829	83434	50
new3	San Jose Mercury[a]	9558	83487	44
tdt	TDT2[c]	9824	55112	193
reu	Reuters-21578[d]	10369	35297	119
oshcal	Ohsumed-233445[a]	11162	11465	10

[a] http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/datasets.tar.gz

[b] http://www.trec.nist.gov

[c] http://www.nist.gov/speech/test/tdt.html

[d] http://www.davidlewis.com/resources/testcollections/

identified the topics in each collection. For reu dataset we only used the stories that have been tagged with the attribute "TOPICS=YES" and include a BODY part.

In our experiments, the documents are represented using the traditional vector space model. Document terms represent the lemmas of the words appearing in the texts (stop words are disregarded) and they are statistically weighted using TF (term frequency in the document).

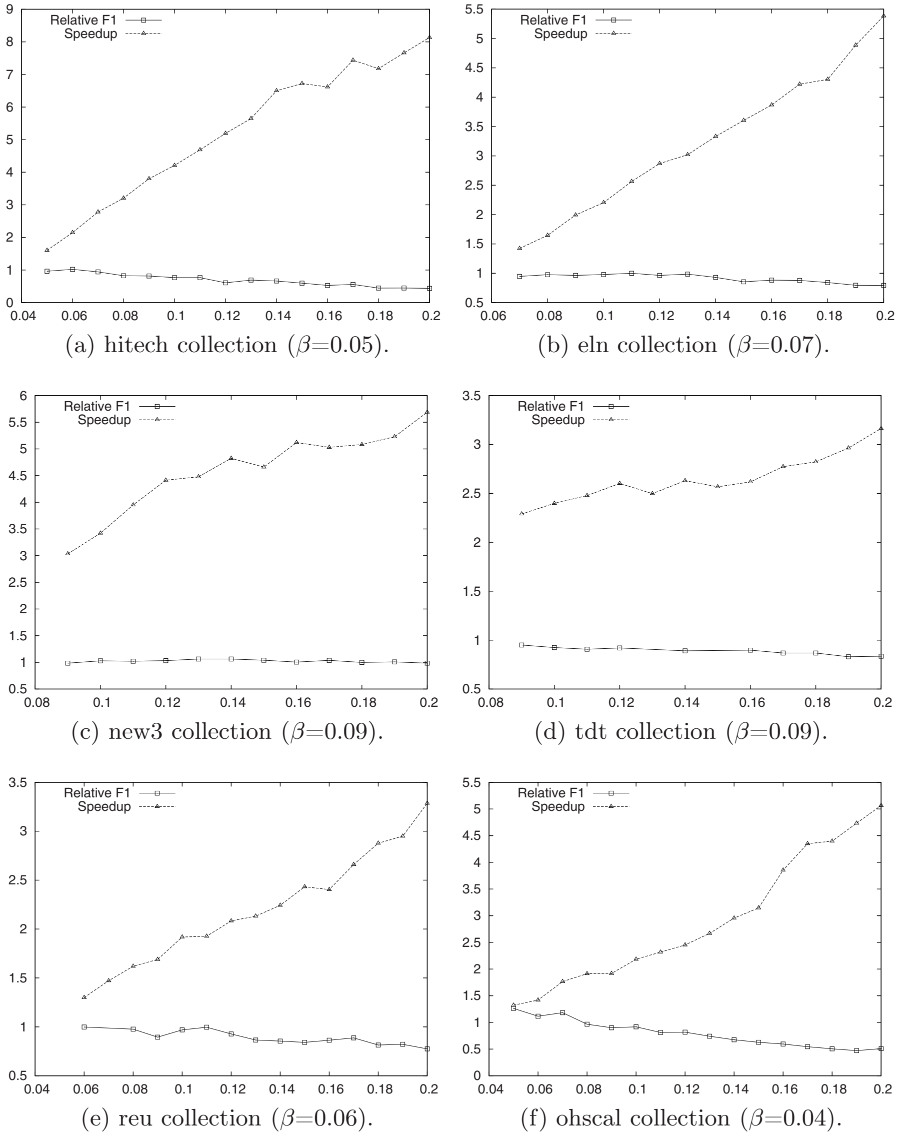
There are several measures to evaluate the quality of hierarchical clustering. We adopt a widely used Overall F1-measure [5], which compares the system-generated clusters at all levels of the hierarchy with the manually labeled topics and combines the precision and recall factors.

The F1-measure of the cluster c_j with respect to the topic t_i can be evaluated as follows:

$$F1(t_i, c_j) = 2\frac{n_{ij}}{n_i + n_j}$$

where n_{ij} is the number of common members in the topic t_i and the cluster c_j, n_i is the cardinality of t_i, and n_j is the cardinality of c_j. To define a global measure, first each topic must be mapped to the cluster that produces the maximum F1-measure:

$$\sigma(t_i) = \arg\max_{c_j} \{F1(t_i, c_j)\}$$



(a) hitech collection (β=0.05).

(b) eln collection (β=0.07).

(c) new3 collection (β=0.09).

(d) tdt collection (β=0.09).

(e) reu collection (β=0.06).

(f) ohscal collection (β=0.04).

Fig. 2. Relative F1 scores and speed-ups obtained by our method

Table 2. Calculated similarities with at most 5% of quality loss

		DHC		*Our Method*				
Collection	β	F1	Calculated Sim.	γ	F1	%	Calculated Sim.	%
hitech	0.05	0.52	4103981	0.06	0.53	102	901467	21
eln	0.07	0.47	24187337	0.13	0.46	99	3779454	16
new3	0.09	0.56	70278443	0.20	0.55	99	5781763	8
tdt	0.09	0.84	75813025	0.09	0.78	95	7270998	10
reu	0.06	0.55	75590891	0.11	0.55	100	14098200	19
oshcal	0.04	0.30	88186250	0.08	0.29	97	13490358	15

Hence, the Overall F1-measure is calculated as follows:

$$Overall\ F1 = \frac{\sum_{i=1}^{N} n_i F1(t_i, \sigma(t_i))}{\sum_{i=1}^{N} n_i}$$

where N is the number of topics. The higher the Overall F1-measure, the better the clustering is, due to the higher accuracy of the clusters mapping to the topics.

The experiments were focused on to compare the proposed version against the original *DHC* algorithm in terms of clustering quality and time efficiency. From the results reported in [1], we choose the parameter β that produces the best hierarchy with respect to Overall F1 measure for each text collection.

To quantitatively compare the relative performance of both methods, we divided the F1 score obtained by the proposed method by the corresponding score obtained by *DHC* using the same (best) β value. We referred to this ratio as *relative F1*. We also calculated the speed-up obtained by the proposed method, that is, the ratio between the execution times of *DHC* and our method.

Figure 2 shows the relative F1 scores, as well as the speed-ups obtained when we vary the γ value from the best β value for each text collection to 0.2. As we can observe, the higher γ value the speed-ups rapidly grow on while the clustering quality slightly decreases. In all text collections, speed-ups of 2-3 can be achieved with less than 5 % loss in clustering quality. Notice also that small speed-ups are obtained with the same F1 score when $\gamma = \beta$.

Table 2 illustrates the number of calculated similarities and the Overall F1 score obtained by the original *DHC* and our method for the best β-value in each document collection. In our method, we select the γ value that produces at least a 95% of clustering quality with respect to that obtained by *DHC*. Columns 7 and 9 represent the percentage of F1 score and the number of calculated similarities obtained by our method with respect to *DHC*, respectively. As we can see, our algorithm significantly reduces the number of calculated similarities. This is the reason why the proposed method achieves good speed-ups.

4 Conclusions

In this paper, a version of *Dynamic Hierarchical Compact* clustering algorithm that improves the updating of the hierarchy has been proposed. Its most important novelty is its ability to profit from the cluster hierarchy previously built by the algorithm to efficiently find out the β-similar clusters to a given one. Exploiting ideas from the access methods, our approach is able to significantly reduce the number of calculated similarities.

The experimental results on several benchmark text collections show that the proposed method is significantly faster than the original *DHC* algorithm while achieving approximately the same clustering quality. Thus, we advocate its use for tasks that require dynamic clustering of large text collections, such as creation of document taxonomies and hierarchical topic detection.

As we showed in the experiments, the accuracy of our algorithm depends on the parameters β and γ, and its best values are different in each text collection. This provides further motivation to study in depth ways for estimating these parameters.

References

1. Gil-García, R.J., Badía-Contelles, J.M., Pons-Porrata, A.: Dynamic Hierarchical Compact Clustering Algorithm. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 302–310. Springer, Heidelberg (2005)
2. Ciaccia, P., Patella, P., Zezula, P.: M-Tree: An efficient access method for similarity search in metric spaces. In: VLDB 1997, pp. 426–435 (1997)
3. Berchtold, S., Bohm, C., Jagadish, H.V., Kriegel, H.P., Sander, J.: Independent quantization: An index compression technique for high dimensional data space. In: 16th International Conference on Data Engineering, pp. 577–588 (2000)
4. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: International Conference on Information and Knowledge Management, pp. 515–524 (2002)
5. Larsen, B., Aone, C.: Fast and Effective Text Mining Using Linear-time Document Clustering. In: KDD 1999, pp. 16–22. ACM Press, New York (1999)

Incorporating Linguistic Information to Statistical Word-Level Alignment*

Eduardo Cendejas, Grettel Barceló, Alexander Gelbukh, and Grigori Sidorov

Center for Computing Research,
National Polytechnic Institute,
Mexico City, Mexico
{ecendejasa07,gbarceloa07}@sagitario.cic.ipn.mx
http://www.gelbukh.com,
http://cic.ipn.mx/~sidorov

Abstract. Parallel texts are enriched by alignment algorithms, thus establishing a relationship between the structures of the implied languages. Depending on the alignment level, the enrichment can be performed on paragraphs, sentences or words, of the expressed content in the source language and its translation. There are two main approaches to perform word-level alignment: statistical or linguistic. Due to the dissimilar grammar rules the languages have, the statistical algorithms usually give lower precision. That is why the development of this type of algorithms is generally aimed at a specific language pair using linguistic techniques. A hybrid alignment system based on the combination of the two traditional approaches is presented in this paper. It provides user-friendly configuration and is adaptable to the computational environment. The system uses linguistic resources and procedures such as identification of cognates, morphological information, syntactic trees, dictionaries, and semantic domains. We show that the system outperforms existing algorithms.

Keywords: Parallel texts, word alignment, linguistic information, dictionary, cognates, semantic domains, morphological information.

1 Introduction

Given a bilingual or multi-lingual corpus, i.e., a set of texts expressing the same meaning in various languages, the text alignment task establishes a correspondence between structures, e.g., words, of the texts in the two languages. For example, given the two texts: English *John loves Mary* and French *Jean aime Marie*, word alignment task consists in establishing the correspondences *John* ↔ *Jean*, *loves* ↔ *aime*, *Mary* ↔ *Marie*.

Text alignment is useful in various areas of natural language processing, such as automatic or computer-aided translation, cross-lingual information retrieval

[*] Work done under partial support of Mexican Government (SNI, CONACYT grant 83270 and 50206-H, and SIP-IPN grant 20090772 and 20091587).

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 387–394, 2009.

and database querying, computational lexicography, contrastive linguistics, terminology, and word sense disambiguation, to mention only a few.

In recent years, many text alignment techniques have been developed [1], [2]. Most of them follow two main approaches: linguistic and statistical or probabilistic [3], [4]. Linguistic approaches use linguistic information, which implies its reliance on availability of linguistic resources for the two languages. Probabilistic approaches involve difficult-to-implement probabilistic generative models. Statistical approaches are simpler since they are based on frequencies of occurrences of words, though they imply high computational cost and usually give lower precision.

In this paper, we present a hybrid alignment algorithm based on a combination of traditional approaches. Its flexibility allows adapting it to the computational environment and to available linguistic resources.

2 Related Work

Ideally, the units (words, sentences, paragraphs) of the two texts ought to be in direct one to one correspondence. However, the alignment task is complicated by many effects that break such an ideal model. One effect is that sometimes the correspondence is not $1 \leftrightarrow 1$ (one word from the source text corresponding to one word in its translation) but $1 \leftrightarrow M$, $M \leftrightarrow 1$, $M \leftrightarrow M$, $1 \leftrightarrow \emptyset$ and $\emptyset \leftrightarrow 1$, where M stands for many words and $\emptyset$ stands for none (empty string). Another effect, specific mainly for the word level, is that the words in the two texts can follow in different order, e.g., English *a difficult problem* vs. Spanish *un problema difícil*.

Most of the alignment systems are oriented on low-inflective languages, for this reason they use wordforms as the basic unit. In the case of highly inflective languages this leads to high data sparseness, rendering statistic translation nearly impossible. For instance, Spanish is a rather highly inflective language, especially in its verbal system, where the complex conjugation produces many wordforms from the same verbal root [5].

It is possible to construct alignment methods based on generative models [6]. Although the standard models can, theoretically, be trained without supervision, in practice several parameters should be optimized using labeled or tagged data. What is more, it is difficult to add characteristics to the standard generative models [7].

Other systems are based on linguistic resources [8], [9]. The use of linguistic resources can present yet another problem for word alignment task. There are two cases as to the use of resources: limited or unlimited [10]. We believe that the more resources are available to the system the better the alignment accuracy. This leads us to the idea of a hybrid combined method for word-level alignment.

This approach is not new. In [11], for instance, a hybrid system is presented, in which the outputs of different existing alignment systems are combined. However, in that approach, interpreting the outputs of the systems is necessary and the user has to define the confidence threshold for each system. De Gispert et al.

proposed in [12] a method to incorporate linguistic knowledge in statistical phrase-based word alignment, but the linguistic information is only used to takes final decisions on unaligned tokens. In [13], parse trees and a few phrase reordering heuristics were incorporated after using the alignment lexicon generated by a statistical word aligner. Another systems have just included morphosyntactic knowledge, as in [14].

3 Alignment Algorithm

The proposed alignment algorithm combines statistical and linguistic approaches. Due to the simplicity of statistical algorithms, approaches of this kind are a starting point for the alignment in our system. Nevertheless, the morphological and syntactical differences between the languages cause multiple errors in such alignment. It is for this reason that, at a later stage, linguistic-based processing is carried out that reinforces or weakens the alignment hypotheses previously obtained with the statistical methods.

3.1 Statistic Processing

There are many well-known statistical alignment methods. Some of them intent to align texts written in very different characters sets, such as English vs. Chinese. The approaches of this paradigm are classified as associative or estimation-based. K-Vec [15] and IBM Models 1 and 2 [16] are examples of associative statistical methods.

The statistical stage of the proposed system relies on three different techniques: (1) Modified K-Vec algorithm, boolean, (2) Modified K-Vec algorithm, with frequencies and (3) IBM Model 2.

Our modified K-Vec algorithm is slightly changed as compared to the original K-Vec presented by Fung & Church [15]. K-Vec algorithm starts with segmentation of the input texts: the texts are divided into small parts and each of the parts is processed independently. The original K-Vec algorithm allows the text to be divided into small pieces or segments. Our modification allows the pieces to be paragraphs, sentences, or a specific number of words (a window). These very convenient division options streamline the statistical process, since its use largely depends on the size of the text segments.

The next step consists in generating a list of words with an associated vector. This vector contains the occurrences of the word in each one of the segments resulting from the division of the text. In the first technique, (modified K-Vec), only boolean values are used to indicate the presence (1) or absence (0) of the word. In the second technique, the frequency of occurrences (i.e., the number of times that the word occurs in a segment) is recorded.

The list of words founded in the text is also used to optimize the later linguistic processing and can also contain the frequency of occurrences of each word in the complete text.

After the list has been completed, the vector corresponding to each word in the source language is compared to all the vectors obtained in the translation,

$$\boxed{\begin{array}{ll}
V(\underline{este}) & = \{0,\underline{1},0,\underline{1},0,0,\underline{1},0,0,0\} \\
 & \\
V(\underline{this}) & = \{0,\underline{1},0,\underline{1},0,0,\underline{1},0,0,0\} \\
V(is) & = \{1,0,0,1,0,1,1,0,1,0\} \\
V(my) & = \{0,0,0,0,0,0,1,1,0,0\} \\
V(\underline{car}) & = \{0,\underline{1},0,\underline{1},0,0,\underline{1},1,0,1\}
\end{array}}$$

Fig. 1. Comparison among vectors

with the purpose of finding those words that match as to their occurrences in each segment. For example, in Fig. 1, the occurrences of the vector corresponding to *este* coincide with those of the words *this* and *car*, with occur in the segments 2, 4 and 7.

Using the correspondences between the vectors of both languages, a contingency table is built to represent information on each pair of related words. Then, the similarity of the pair is calculated for each table. The similarity of words is determined by means of an association test. Our system incorporates the following similarity measures: – Pointwise Mutual Information (PMI), – T-score, – Log-likelihood ratio, and – Dice coefficient.

After all association values have been calculated, the word with the greatest level of association is selected and the other candidates are discarded. In this way, a dictionary is created from the translation words that better correspond to each source word. If the algorithm is used in a bidirectional way, then the same process is carried out interchanging the source and target languages [17] and the best averages of both results are obtained to acquire the best word pairs.

If the algorithm does not use linguistic information, then after this stage a file of final alignments is created, indentifying each word and its position in both texts.

3.2 Linguistic Processing

The methods developed following the linguistic approach make use of diverse resources, such as bilingual dictionaries [16], lexicons with morphological information [9] and syntactic trees [8]. In addition to these, in our algorithm we incorporate the use of semantic domains.

Dictionaries allow for extraction of lexical information. In this way, the word from the source text is considered along with all its possible translations in the target text. These data can then be employed in the calculation or adjustment of the probabilities of the correspondences obtained in the statistical phase.

Similarly, the morphological and syntactical information are knowledge sources useful for increasing or decreasing the certainty of each alignment hypothesis. Using morphological information, it is possible to compare lemmas and verify grammatical categories [9]. On the other hand, knowing the syntax of the sentences allows the identification of its parts (subject, predicate, etc.) and facilitates comparisons with its counterparts in the target language.

Finally, semantic domains provide a natural way to establish semantic relations between the meanings of a word. Roughly speaking, the method consists of rejecting those translations that lay in different domains from the original word, and giving greater weight to those that lay in the same domains. We used WordNet Domains [18] to extract the domain labels. This is a rarely used concept, utilized to locate or train the aligner in a specific domain [19].

In addition to the above linguistic resources, we use a heuristic of cognates. Shared sequences of characters are looked in both texts, for example: English *organization* and Spanish *organización*. In this way it is easier to align words that totally or partially coincide (for example, proper nouns). The minimum percentage of coinciding letters in the two words to consider them as cognates is a user-defined parameter of the system. False cognates are taken into account by using a predefined list of known false cognates.

Unlike most of the alignment models, where training is carried out with the EM (Expectation Maximisation) algorithm [20], our system allows using previous alignments that can be difficult to find. All the alignment hypotheses that can be obtained with different methods will serve as a reference for future alignment tasks. It is important to mention that the system can start the analysis using purely linguistic data similar to some proposed methods [21], if it is configured to do so.

4 General Architecture

In order to test the proposed ideas, we have implemented a highly configurable and flexible system that we called HWA (Hybrid Word Aligner), which allows our method to be adapted to the implementation environment and the availability of resources. The combination of the statistical and linguistic approaches has the purpose of obtaining a parameterizable algorithm that can be used in different ways depending on the requirements of the expected results.

The architecture of the alignment system is shown in Fig. 2. The alignment process is subdivided into three main phases: preprocessing, main algorithm, and output generation.

5 Preliminary Results

While statistic processing can be applied to any language pairs, the linguistic processing module requires specific grammatical resources, with the exception of the cognate detection[2] and the learning information. We have chosen for our experiments Spanish–English parallel texts from five novels (*Dracula*, *Don Quixote of la Mancha*, *The Shop of Ghosts*, *Little Red Riding Hood* and *The Haunted House*). The selection of fragments was made randomly by the paragraph number. It is important to emphasize that in every case the paragraphs

[2] Providing the languages have the same type of characters.

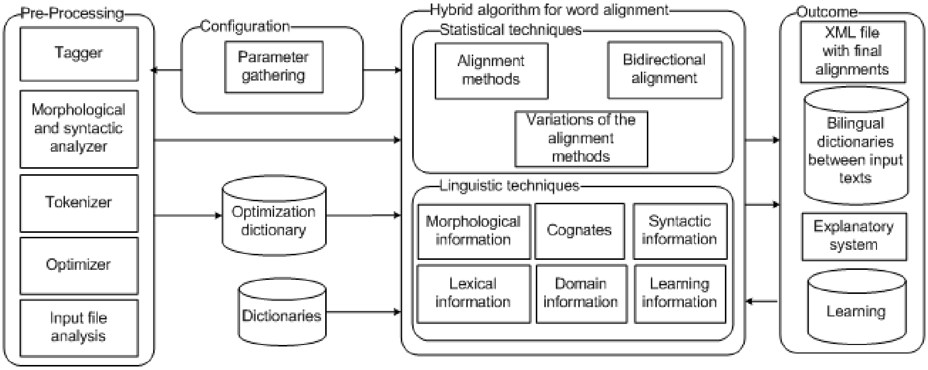


Fig. 2. General architecture of the aligner

Table 1. Results

Aligner	% successes	generation of dictionary?	generation of alignment file?
GIZA++	52	Yes	Yes
Uplug	45	No (not as a result)	Yes
K-Vec++	35	Yes (not for all the words)	No
HWA	53	Yes	Yes

have been previously aligned at sentence level. While this is required for other systems but not for our system, yet this influences the alignment quality.

Table 1 shows the obtained results in terms of precision: the percentage of established correct alignment correspondences. We compare our results with those of three other aligners: GIZA++ [22], original K-Vec [23], and Uplug [24]. The results of the proposed algorithm (HWA) were obtained by executing the modified K-Vec procedure during the statistical processing and by applying the cognates during the linguistic processing to reinforce the alignments. For the moment, we did not apply other linguistic modules, that is why we call the obtained results "preliminary".

To obtain the results, each aligner was provided with the parallel texts of the specified novels. The output of each aligner was manually verified to determine the correct alignments, and an average percentage of the correct alignments was obtained for the five input data sets. Due to the differences in the aligners, similar parameters were used in each test: – input parallel texts were the same for each alignment program, – texts had no labels, – no previous training was applied, – learning bases were not applied, – bidirectional alignments were not performed, – text segmentation was performed with the best consideration of each algorithm, and – the same association test was used. It is important to note that the results from the aligners can vary depending on the configuration parameters and on the size of the input texts.

6 Conclusions

We have presented an alignment algorithm that combines two main approaches, statistical and linguistic-based, in order to improve the alignment between words of bilingual parallel texts. We conducted experiments with short texts fragments. The results obtained by the proposed algorithm are better than those of the existing alignment algorithms.

The statistical techniques provides a good starting point for the alignment processes; however, incorporation of linguistic techniques increases the efficiency of the system by involving intrinsic characteristics of the implied languages. The main disadvantage of linguistic processing is the need in the linguistic resources given their limited availability. In addition, this has an impact on the algorithm speed. Nonetheless, employing databases of optimization has proven to minimize this disadvantage. The cost-benefit trade-off of linguistic techniques implies a great emphasis on the particular configuration of the algorithm so as to obtain the best alignments. This is due to the fact that the system allows free incorporation or exclusion of linguistic resources at any given moment during the process.

Combining statistical and linguistic techniques is a viable option thanks to the current computing capacities and will be more acceptable as the speed of computers grows, costs of hardware (memory and storage) decreases, and more resources become available to natural language processing community.

References

1. Langlais, P., Simard, M., Vronis, J.: Methods and practical issues in evaluating alignment techniques. In: Proceedings of the 17th International Conference on Computational Linguistics, Montréal, pp. 711–717 (1998)
2. Veronis, J.: Parallel Text Processing: Alignment and Use of Translation Corpora. Kluwer Academic Publishers, Dordrecht (2001)
3. McEnery, T., Xiao, R., Tonio, Y.: Corpus-based language studies: An advanced resource book. Routledge, London (2006)
4. Kit, C., Webster, J., Kui, K., Pan, H., Li, H.: Clause alignment for hong kong legal texts: A lexical-based approach. International Journal of Corpus Linguistics 9, 29–51 (2004)
5. Agirre, E., Díaz de Ilarraza, A., Labaka, G., Sarasola, K.: Uso de información morfológica en el alineamiento español-euskara. In: Actas del XXII Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural, Zaragoza, pp. 257–264 (2006)
6. Dale, R., Moisl, H., Somers, H.L.: Handbook of natural language processing. Marcel Dekker Inc., New York (2000)
7. Moore, R.: A discriminative framework for bilingual word alignment. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, pp. 81–88 (2005)
8. Ma, Y., Ozdowska, S., Sun, Y., Way, A.: Improving word alignment using syntactic dependencies. In: Proceedings of the ACL 2008:HLT Second Workshop on Syntax and Structure in Statistical Translation, Ohio, pp. 69–77 (2008)

9. Pianta, E., Bentivogli, L.: Knowledge intensive word alignment with knowa. In: Proceedings of the 20th International Conference on Computational Linguistics, Geneva, pp. 1086–1092 (2004)
10. Mihalca, R., Pedersen, T.: An evaluation exercise for word alignment. In: Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: data driven machine translation and beyond, Edmonton, vol. 3, pp. 1–10 (2003)
11. Ayan, N., Borr, B., Habash, N.: Multi-align: Combining linguistic and statistical techniques to improve alignments for adaptable mt. In: Proceedings of the 6th Conference of the Association for Machine Translation in the Americas, Washington DC, pp. 17–26 (2004)
12. De Gispert, A., Mario, J., Crego, J.: Linguistic knowledge in statistical phrase-based word alignment. Natural Language Engineering 12, 91–108 (2006)
13. Hermjakob, U.: Improved word alignment with statistics and linguistic heuristics. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 229–237 (2009)
14. Hwang, Y., Finch, A., Sasaki, Y.: Improving statistical machine translation using shallow linguistic knowledge. Computer Speech and Language 21, 350–372 (2007)
15. Fung, P., Church, K.: K-vec: A new approach for aligning parallel text. In: Proceedings of the 15th Conference on Computational Linguistics, Kyoto, vol. 2, pp. 1096–1102 (1994)
16. Och, F., Ney, H.: A systematic comparison of various statistical alignment models. In: Proceedings of the 18th Conference on Computational Linguistics, vol. 2, pp. 19–51 (2003); Computational Linguistics 29(1), 19–51 (2003)
17. Tiedeman, J.: Word to word alignment strategies. In: Proceedings of the 20th International Conference on Computational Linguistics, Geneva, pp. 221–218 (2004)
18. Bentivogli, L., Forner, P., Magnini, B., Pianta, E.: Revising wordnet domains hierarchy: Semantics, coverage, and balancing. In: Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources, Geneva, pp. 101–108 (2004)
19. Wu, H., Wang, H., Liu, Z.: Alignment model adaptation for domain-specific word alignment. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Michigan, pp. 467–474 (2005)
20. Och, F., Ney, H.: Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, pp. 440–447 (2000)
21. De Gispert, A., Mario, J., Crego, J.: Phrase-based alignment combining corpus cooccurrences and linguistic knowledge. In: Proceedings of the International Workshop on Spoken Language Translation, Kyoto, pp. 85–90 (2004)
22. GIZA++: Training of statistical translation models,
    http://www.fjoch.com/GIZA++html
23. K-Vec++: Approach for finding word correspondences,
    http://www.d.umn.edu/tpederse/Code/Readme.K-vec++.v02.txt
24. Uplug: The home page, http://stp.ling.uu.se/~joerg/uplug/

VIII Keynote 3

When Pyramids Learned Walking[*]

Walter G. Kropatsch

PRIP, Vienna University of Technology, Austria
krw@prip.tuwien.ac.at
http://www.prip.tuwien.ac.at/

Abstract. A temporal image sequence increases the dimension of the data by simply stacking images above each other. This further raises the computational complexity of the processes. The typical content of a pixel or a voxel is its grey or color value. With some processing, features and fitted model parameters are added. In a pyramid these values are repeatedly summarized in the stack of images or image descriptions with a constant factor of reduction. From this derives their efficiency of allowing log(diameter) complexity for global information transmission. Content propagates bottom-up by reduction functions like inheritance or filters. Content propagates top-down by expansion functions like interpolation or projection. Moving objects occlude different parts of the image background. Computing one pyramid per frame needs lots of bottom-up computation and very complex and time consuming updating. In the new concept we propose one pyramid per object and one pyramid for the background. The connection between both is established by coordinates that are coded in the pyramidal cells much like in a Laplacian pyramid or a wavelet. We envision that this code will be stored in each cell and will be invariant to the basic movements of the object. All the information about position and orientation of the object is concentrated in the apex. New positions are calculated for the apex and can be accurately reconstructed for every cell in a top-down process. At the new pixel locations the expected content can be verified by comparing it with the actual image frame.

1 Introduction

Humans and animals are able to delineate, detect and recognize objects in complex scenes very rapidly. One of the most valuable and critical resources in human visual processing is time. Therefore a highly parallel model is the biological answer to deal satisfactorily with this resource [1]. Tsotsos [2] showed that hierarchical internal representation and hierarchical processing are the credible approach to deal with space and performance constraints, observed in human visual systems. Moreover, Tsotsos [3] concludes that in addition to spatial parallelization, a **hierarchical organization** is among the most important features of the human visual systems.

[*] Supported by the Austrian Science Fund under grants S9103-N13, P20134-N13 and P18716-N13.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 397–414, 2009.

It is now accepted that the human visual system has a hierarchical (pyramidal) architecture and that the visual mechanisms can be adequately modeled by hierarchical algorithms [4]. Pyramid algorithms are adequate models for the Gestalt rules of perceptual organization such as proximity, good continuation, etc. [5,6]. Moreover, Privitera et al. [7] showed, in a stimulation of the human visual system, that there are two strategies to obtain and apply information about the importance of different regions of an image: the **bottom-up** methods retrieve features only from the input image, and **top-down** methods are driven by available knowledge about the world. Thus the hierarchical structure must allow the transformation of **local** information (based on sub-images) into **global** information (based on the whole image), and be able to handle both locally distributed and globally centralized information. This data structure is known as **hierarchical architecture** or **pyramid** [8].

The (image) pyramid might be the answer to the time and space complexity in computer vision systems, by implementing both processing strategies: bottom-up and top-down. This hierarchical structure allows distribution of the global information to be used by local processes. The main advantage of the hierarchical structures is rapid computation of a global information in a recursive manner. The change of local over to global information, e.g. from pixels arrays to descriptive data structures, is a point of discontinuity in vision systems [8]. Hierarchical structures offer a way to alleviate this discontinuity, where global structures become local in higher levels of this hierarchy.

1.1 Recall on Image Pyramids

Tanimoto [9] defines a **pyramid** as **a collection of images of a single scene at different resolutions**. In the *classical pyramid* every 2×2 block of cells is merged recursively into one cell of the lower resolution. We formally describe this structure by $2 \times 2/4$ which specifies the 2×2 reduction window and the reduction factor of 4. This type of pyramid has been extensively studied (e.g. [10], [11]).

Tanimoto's formal definitions refer to this type of pyramid [12]. He defines a cell (Tanimoto uses the term *pixel*) in a pyramid as a triple (x, y, v) which is defined in a **hierarchical domain** of n levels:

$$\{(x, y, v) | 0 \leq x \leq 2^v, 0 \leq y \leq 2^v, 0 \leq v \leq n-1\} \tag{1}$$

Then a pyramid is any function whose domain is a hierarchical domain. This function assigns to every cell in the simplest case a value, but also structures of higher complexity can be stored.

1.2 The Flow of Information within a Pyramid

Information necessary to connect the observed part of the object with the parts in the adjacent cells must be passed up to the next lower resolution level (or equivalently, to the next higher pyramid level). There, the cells cover a larger area and can join some parts of the level below. This process is repeated up to successively lower resolutions until the whole object is within the observation window of a cell. Unfortunately, in some pyramid structures a small rigid motion (shift, rotation) of the object may cause a completely different representation (the representation cell may be many levels below or above. [13]). This problem is resolved by the adaptive pyramid [14] which is the direct precursor of the irregular pyramid (see section 3).

An important class of operations is responsible for the bottom-up information flow within the pyramid: the **reduction function** R. It computes the new value of a cell exclusively from the contents of its children. Given an image in the base of the pyramid, application of a reduction function (e.g. average) to all first level cells fills this level. Once the cell of the first level received a value, the same process can be repeated to fill the second level and so on to the top cell. With these operations the levels $G_i, i = 0, \ldots, n$ of a (Gaussian) pyramid are generated by following iterative process: $G_0 := I; \quad G_{i+1} \ := R(G_i), i = 0, \ldots, n-1$.

1.3 Laplacian Image Pyramids

Burt [15] describes a method for compressing, storing and transmitting images in a computationally efficient way.

Let G_k denote a $5 \times 5/4$ Gaussian pyramid, where k denotes the different levels and G_0 is the base. The bottom-up building process is based on the reduction function R: $G_k := R(G_{k-1})\,,\ k := 1, 2, \ldots n$. The reduction function maps the children's collective content into the properties of the parent.

The Gaussian smoothing filter has a low-pass characteristic removing only the highest frequencies. Therefore the Gaussian pyramid $G_k; k = 0, \ldots, n$ contains a high amount of redundancy which is substantially reduced in the Laplacian pyramid:

1. The expansion function E is the reverse function of the reduction function R. It expands (interpolates) the properties of the parent(s) cells into the children's content at the higher resolution level.
2. The 'reduce - expand' RE Laplacian pyramid compares the child's content with the expanded content of the parents and simply stores the difference:

$$L_l := G_l - E(G_{l+1}) \text{ for } l := 0, 1, \ldots, n-1 \tag{2}$$

3. Reconstruction of G_k is exact: $G_k := L_k + E(G_{k+1})$ for $k := n-1, n-2, \ldots, 0$.
4. Hence storing $G_n, L_{n-1}, L_{n-2}, \ldots, L_0$ is sufficient for exact reconstruction of the original image G_0.

Note that the intensity of the reconstructed image depends on the intensity of the apex. *If the grey value of the apex is increased the intensity of the whole reconstructed image is increased by the same value.* We observe that all the levels below the apex of the Laplacian pyramid *are invariant to global changes in intensity.*

1.4 First Steps in a Dynamic World

In [16], the Laplacian pyramid has been used to indicate a significant change in a time-series of images. Let $I(t)$ denote the image taken at time t, let m denote the level at which the change shall occur. Following procedure initiates an **alarm** when an unusual situation occurs in the field of view:

1. $D(t) := I(t) - I(t-1)$;
2. build Laplacian pyramid $L_i(t), i := 1, 2, \ldots, m$ with $L_0(t) := D(t)$;
3. square level m: $L_m(t)^2$;
4. build Gaussian pyramid $G_k(t), k := 1, 2, \ldots, n$ with $G_0(t) := L_m(t)^2$;
5. threshold $G_k(t), k := 1, 2, \ldots, n$: alarm.

In this case the base of the Laplacian pyramid are the frame differences. It is computed bottom-up up to level m which identifies the frequency band at which the event causes the alarm. This nicely eliminates high frequency components and false-alarms caused by noise or tree branches moving in the wind.

Although this early use of pyramids for detecting dynamic changes in an image sequence was used in several applications it focused on a single event and could not filter out a description of the alarm causing event.

1.5 Some Words on Graphs

Graph hierarchies allow to use other spatial orderings of image primitives, not only the regular spatial structures like arrays. Image primitives (e.g. pixels, edges, etc.) are represented by vertices and their relations by edges of the graph. These vertices and edges are attributed. A classical example of graph representation of a set of primitives is the **region adjacency graph** (RAG), where each image region is represented by a vertex, and adjacent regions are connected by an edge. Attributes of vertices can be region area, average gray value, region statistics etc.; and attributes of edges can be the length of the boundary, the curvature, etc. between the pair of adjacent regions. The graph hierarchy is then built by aggregating these primitives. The main application area of the region based representation is *image segmentation* and *object recognition* [17]. Note that region adjacency graph (RAG) representation is capable to encode only the neighborhood relations.

1.6 And Some Words on Image Segmentation

An image segmentation partitions the image plane into segments that satisfy certain homogeneity criteria (see [18] for an overview). There are many reasons for using the hierarchical paradigm in image partitioning [19]:

- the scale at which interesting structure is important is not known in advance, therefore a **hierarchical image representation** is needed;
- **efficiency of computation**: the results obtained from the coarse representation are used to constrain the costly computation in finer representations; and

- **bridging the gap** between elementary descriptive elements (e.g. pixels) and more global descriptive elements, e.g. regions (see [20]).

Although the goal of image segmentation is producing a single partition of the image, and not necessarily a hierarchy, the hierarchical representation is needed, especially if the image context is not taken into consideration. The idea behind this is if you do not know what you are looking for in an image, then use a hierarchical representation of the image, and moreover a data structure that allows the ability to access the finest partitioning (in our case the bottom of the pyramid) or in case of 'bad' partitioning the faculty to repair these 'errors'. A wide range of computational vision tasks could make use of segmented images, just to mention some: object recognition, image indexing, video representation by regions etc., where such a segmentation relies on efficient computation.

1.7 Overview of the Paper

After discussing current representations of objects with both spatial and temporal structure (like articulation), we recall the basic concept of irregular graph pyramids in Section 3. Their basic properties are then efficiently applied in the new concept for describing the temporal evolution of a tracked object (Section 4). It relates the principles of the Laplacian pyramid with the graph pyramid to separate two types of information: the trajectory and the dynamic orientation is concentrated in the apex of the object (*only one 'foot' is updated at each step*), while all the lower levels code the spatial structure of the object if it is rigid (Section 5). Extensions lossless rotation, articulated parts and adaptive zoom are shortly addressed in Section 6. The conclusion (Section 7) summarizes the major advantages of the new proposal and lists some of the many future applications of the concept.

2 Objects with Structure in Space and in Time

In physics, motion means a change in the location of a "physical body" or parts of it. Frequently the motion of a (mathematical/geometrical) point is used to represent the motion of the whole body. However in certain cases (e.g. parking a car) more information than a single point is required. Because describing an object by an un-ordered set of all its points and their motion is not optimal (considering for example storage space, redundancy, and robustness with respect to missing or incorrect information), we can use the part structure of natural physical bodies (e.g. "objects") to represent them in a more efficient way.

In the context of computer vision, a representation for an object can be used to model knowledge (e.g. appearance, structure, geometry) about the object and its relation to the environment. This knowledge can be used for tasks like: verifying if a certain part of an image is the object of interest, identifying invalid configurations, guiding the search algorithm for a solution/goal, etc. These tasks are in turn used by processes like segmentation, tracking, detection, recognition, etc.

Considering representations for structured objects, we identify the following spatial and temporal scales. On the **spatial scale** there are representations considering:

i. no spatial decomposition information;
ii. statistical information about the parts (e.g. number of parts/features of different type);
iii. "static" structure i.e. adjacency of parts;
iv. degrees of freedom (e.g. articulation points);
v. pose relations between parts – correct/incorrect configurations/poses.

On the **temporal scale** we have:

a. no temporal information;
b. instant motion (e.g. speed and direction at a certain time instance);
c. elementary movement (e.g. moving the arm down);
d. action (e.g. a step, serving in tennis);
e. activity (e.g. running, walking, sleeping, playing tennis).

On the spatial scale, representations cover the whole domain from i. to v. (see [21,22,23,24]). There are simple representations like *points* [25,26,27], *geometric shapes* (rectangle, ellipse) [28,29], and *contours/silhouettes* [30,31], but also more complex ones [32,33]. Felzenszwalb et al. [32] use pictorial structures to estimate 2D body part configurations from image sequences. Navaratnam et al. [33] combine a hierarchical kinematic model with a bottom up part detection to recover the 3D upper-body pose. In [34] a model of a hand with all degrees of freedom and possible poses is used.

On the temporal domain, most methods use simple motion models, typically considering the motion between a few consecutive frames. More complex representation on the temporal domain can be found in *behavior understanding*, where dynamic time warping (e.g. [35]), finite-state machines (e.g. [36]), and hidden Markov models (e.g. [37,38]) are employed. In the fields of pose estimation and action recognition there is a so-called state space representation. For example a human can be represented by a number of sticks connected by joints [39]. Every degree of freedom of this model is represented by an axis in the state space. One pose of a human body is one point in this high-dimensional space and an event/action is a trajectory in this space. This trajectory through the state space is one possibility to represent the temporal aspect [40]. Nevertheless, there are still very few works that look at complex spatial and temporal structure at the same time (e.g topology in the 4D spatio-temporal domain).

In the context of computer vision, properties relating the objects with the visual input also need to be represented. Considering the **dynamics of a description** created using a certain representation, one can look at how *a description* and its *building/adapting processes* behave, when the represented information changes. For example: number of parts or their type, static structure, type of activity, relation to visual input (scaling, orientation), etc.

For small changes in the information a minimal change in the description is desired. E.g. scaling, rotation, part articulation, illumination, should only minimally affect the description.

In addition to the dynamics, one can talk about the **genericness** of a representation i.e. the ability to represent objects of varying degree of complexity and abstraction (e.g. industrial robot, normal human walking, stone).

3 Irregular Graph Pyramids

Pyramids can be built also on graphs. In this case the domain is no more the simple array structure as in Tanimoto's definition but a graph where the function values are stored as attributes of the vertices of the graph. A RAG encodes the adjacency of regions in a partition. In the simplest case a vertex corresponds to a pixel and the edges encode the 4-neighborhood relations (Fig. 1). The dual vertices correspond in this case to the centers of all 2×2 blocks, the dual edges are the cracks between adjacent pixels. More generally, a vertex can be associated to a region, vertices of neighboring regions are connected by an edge. Classical RAGs do not contain any self-loops nor parallel edges. An *extended region adjacency graph* (eRAG) is a RAG that contains some *pseudo edges.* Pseudo edges are the self-loops and parallel edges that are required to encode neighborhood relations to a cell *completely enclosed* by one or more other cells [41] i.e. they are required to correctly encode the topology. The *dual* graph of an eRAG G is called the *boundary graph* (BG, see Fig. 2) and is denoted by $\bar{G}$. The edges of $\bar{G}$ represent the boundaries (borders) of the regions encoded by G, and the vertices of $\bar{G}$ represent points where boundary segments meet. G and $\bar{G}$ are planar graphs. There is a one-to-one correspondence between the edges of G and the edges of $\bar{G}$, which also induces a one-to-one correspondence between the vertices

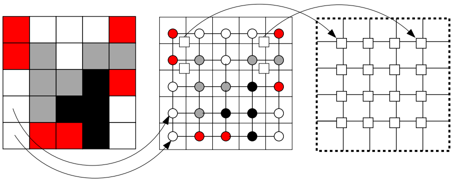


Fig. 1. Image to primal and dual graphs

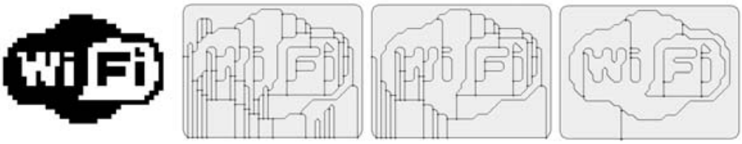


Fig. 2. A digital image I, and boundary graphs $\bar{G}_6$, $\bar{G}_{10}$ and $\bar{G}_{16}$ of the pyramid of I

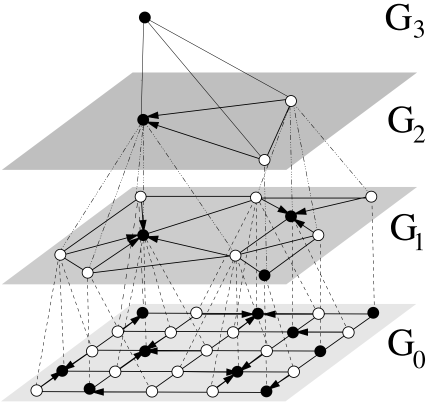


Fig. 3. Example graph pyramid

of G and the 2D cells (will be denoted by *faces*[1]) of $\bar{G}$. The dual of $\bar{G}$ is again G. The following operations are equivalent: edge contraction in G with edge removal in $\bar{G}$, and edge removal in G with edge contraction in $\bar{G}$.

A (dual) irregular graph pyramid [41,42] is a stack of successively reduced planar graphs $P = \{(G_0, \bar{G}_0), \ldots, (G_n, \bar{G}_n)\}$ (Fig. 3). Each level $(G_k, \bar{G}_k), 0 < k \leq n$ is obtained by first contracting edges in G_{k-1} (removal in $\bar{G}_{k-1}$), if their end vertices have the same label (regions should be merged), and then removing edges in G_{k-1} (contraction in $\bar{G}_{k-1}$) to simplify the structure. The contracted and removed edges are said to be *contracted* or *removed* in $(G_{k-1}, \bar{G}_{k-1})$. In each G_{k-1} and $\bar{G}_{k-1}$ the contracted edges form trees called *contraction kernels*. One vertex of each contraction kernel is called a *surviving vertex* and is considered to have 'survived' to $(G_k, \bar{G}_k)$. The vertices of a contraction kernel in level $k-1$ form the *reduction window* of the respective surviving vertex v in level k. The *receptive field* of v is the (connected) set of vertices from level 0 that have been 'merged' to v over levels $0 \ldots k$.

4 Moving Objects

The study of dynamic image sequences (or videos) aims at identifying objects in the observed image sequence and describing their integrated properties and their dynamic behaviour. There are several possibilities to segment an object from an image or a video:

- image segmentation methods are able to locate image regions in individual images that are 'homogeneous' in certain terms. Examples are David Lowe's SIFT-features [43], different variants of Ncut [44] or the MST pyramid [45]. Objects of interest are, however, mostly composed of several such regions and further grouping is required.

[1] Not to be confused with the vertices of the dual of a RAG (sometimes also denoted by the term *faces*).

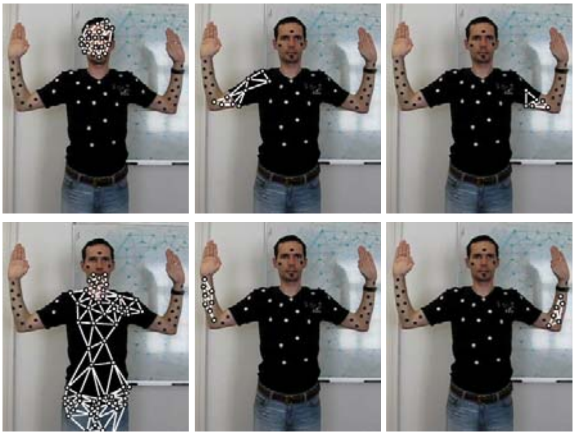


Fig. 4. Example of an extracted object and its rigid parts

- Optical flow approaches overcome the grouping since the different parts of an object usually move together.
- detection of interest points and tracking them individually over the sequence. In order to preserve the structure of points belonging to the same object pairwise relations like distances can be used efficiently to overcome failures caused by noise or occlusions (see [46,47]).

4.1 Extraction of Structure from Videos

In [47] a graph-pyramid is used to extract a moving articulated object from a video, and identify its rigid parts. First a spatio-temporal selection is performed, where the spatial relationships of tracked interest points over time are analysed and a triangulation is produced, with triangles labeled as *potentially-rigid* and *non-rigid*. The *potentially-rigid* triangles are given as input to a grouping process that creates a graph pyramid such that the each top level vertex represents a rigid part in the scene. The orientation variation of the input triangles controls the construction process and is used to compute the similarity between two regions. This concept is related to the single image segmentation problem [17], where the results should be regions with homogeneous color/texture (small internal contrast) neighbored to regions that look very different (high external contrast). In our case the “contrast” is interpreted as the inverse of “rigidity”. The result of this method can be used to initialize an articulated object tracker. Fig. 4 shows an example.

4.2 Describing the Tracking Results

Most of the current approaches describe the results in the domain of the original data and use the image and frame coordinates. The resulting trajectory consists in a sequence of frame coordinates where the object was at the respective time

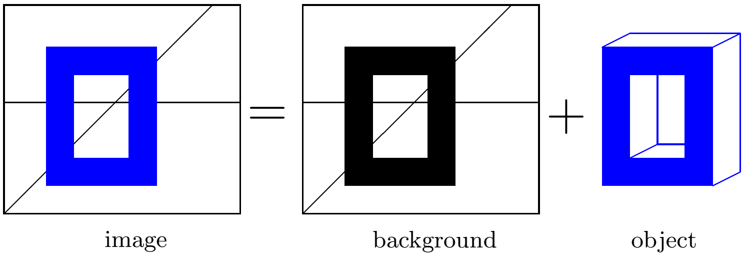


Fig. 5. Separating the object from the background

instance. We consider the use of a separate data structure for each moving object in order to update independent movements and properties in clearly separated data (i.e. to describe 'walking').

Once an object is identified in an image (frame) or even in an image pyramid we cut out the object from its pixel based representation into the neighborhood graph of pixels, close its surface topologically by invisible surface patches of the backside (Fig. 5). The remaining image is considered as background and the pixels of the removed object are labelled as invisible.

4.3 Topological Completion: Represent a 3D Object

In a video frame, a 3D object may be occluded or partially visible. We call the visible part of the surface the *front surface*. From a single image frame, the front surface is extracted as a graph. This extracted graph embeds the topological structure and discriminative visible features of the object. In the vertex, attributes like size, color and position of the corresponding pixels (region) can be stored and the edges specify the spatial relationships (adjacency, border) between the vertices (regions)[46]. Topological completion closes the visible surface by one or more invisible surface patches in order to completely cover the surface of the volumetric object.

Each level of the irregular graph pyramid is a graph, presenting the closed surface of the moving object in multiple resolutions. We collect the topological structures from the visible surface of the target object. Each graph embeds both features and structural information. Locally, features describe the object details; globally, the relations between features encode the object structure.

For initialization, the base graph of the pyramid encodes the initial information about the object, the graph is closed on the invisible backside to create a closed 2D manifold. The graph pyramid can cope with this structure and the same operations can be applied as in the case of an image. As new visible parts of the surface would reveal previously invisible parts, the object representation is incrementally updated automatically from observing the target object in a video sequence. This requires the registration of the visible parts and the replacement of some invisible patches. When some hidden structure appears, we add the new topological structure into the previous 2D manifold to obtain the updated object representation. For instance, a rotating cup will reveal the handle that was hidden before, and hide the logo when it moves out of sight.

When the camera has covered all the aspects of the object, which means all the observable parts of the object have been integrated in the object model, the topological structure of the target object is complete. This is the process we defined as *topological completion.*

5 Walking: Only One Foot Leaves Contact with the Ground

In the image frame every pixel establishes a contact between the moving object and the digital image. In order to reduce efforts of updating large amounts of data (e.g. geometrically transforming the object window) we reduce the contact to a single point which serves as a reference similar to *the foot making the next step in walking.*

5.1 Invariance to Translation

In order to keep the geometric information of the object's surface patches we attribute each cell $v \in V$ with the coordinates of the corresponding image pixels, $p(v) = (x, y) \in [0, N_x] \times [0, N_y]$. These coordinates could, if necessary, be enhanced by depth values, $p(v) = (x, y, d) \in [0, N_x] \times [0, N_y] \times \mathbb{R}$, coming from different *'shape from X'* methods (e.g. [48]).

Both the extracted objects and the remaining background image can be embedded in an irregular graph pyramid either

- by using the existing image pyramid (e.g. after segmentation) or
- by rebuilding the pyramids of the objects and the background.

The coordinates of the higher level cells can be computed from the children either by inheritance from the surviving child to the parent or by a weighted average of the children's coordinates or by a combination with the selection of survivors such that the largest region survives and inherits its children's coordinates in the case the pyramid is rebuilt. After this bottom-up propagation each cell has 2D or 3D coordinates.

The resulting position attributes $p(v)$ are as redundant as the grey values of a Gaussian pyramid. Hence the idea of expanding the parent's coordinates $p(v_p)$ to the children, $p(c)$, parent$(c) = v_p$, and storing simply the difference vector $d(c) = p(c) - E(p(v_p))$ between the expansion and the original attribute in analogy to the Laplacian pyramid[2]. Let us call the difference $d(c)$ the child's **correction vector**. Similar to the Laplacian pyramid the original position of each cell can be reconstructed accurately (up to numerical precision) by adding all the correction vectors (following the equivalence $p(c) = E(p(v_p)) + d(c)$) up to the apex (a sort of equivalent correction vector). The position of the

[2] In the simplest case, expand by projection, $E(x) = x$.

cell is then the position of the apex added to the sum of correction vectors $p(c_0) = p(\text{apex}) + \sum_{c=c_0, parent(c_0),\ldots}^{\text{apex}} d(c)$. As a side effect the object can be rigidly shifted by simply translating the apex to the desired position and reconstructing the coordinates of all the other cells if needed. This shift invariance of the lower pyramid levels allows simple modifications and efficient reconstruction but needs further adaptation in order to cope with rotation and scale changes.

5.2 Invariance to Rotation and Scale

So far the position of an object is coded in the coordinates $p(\text{apex})$ of the apex. Every cell below the apex contains correction vectors $d(c)$ allowing accurate reconstruction of its position by top-down refinement using the correction vectors.

Most objects have an orientation $o \in \mathbb{R}^3$ in addition to their position $p \in \mathbb{R}^3$. Orientation can be derived from properties like symmetry, moving direction or can be given by the object model a priori. Since orientation is a global property of an object we add it to the properties of the apex of the object's pyramid. The vector $o(\text{apex})$ codes both the orientation with respect to the reference coordinate system and a scale if the length $||o|| \neq 1$ is different from unit length. Orientation and position allow to quickly transform the object pyramid from one coordinate system to another (i.e. of another camera or viewpoint).

The orientation of the object can be used to make correction vectors invariant to rotation and scale. Taking $p(v_p)$ as the position where both the orientation vector and the correction vector start we can express the correction vector $d(c)$ as a rotated and scaled version of the orientation: $d(c) = \lambda R_x(\alpha) R_y(\beta) R_z(\gamma) o$ and store the parameters $r(c) = (\lambda, \alpha, \beta, \gamma)$ as new parameters of the cell c. The angles α, β, γ can be the Euler angles of the corresponding rigid body and the scale factor $\lambda = ||d(c)||/||o||$ relates the vectors' lengths. Given the position $p(v_p)$ of the parent and the orientation o of the object each cell can accurately reconstruct it's position $p(c) = p(v_p) + \lambda R_x(\alpha) R_y(\beta) R_z(\gamma) o$. We note that in addition to the invariance with respect to translation, the parameters $r(c)$ are invariant also to rotation and scale. The rotation of the object is executed by applying the rotation to the orientation of the apex and similar with a scale change.

All the vertices can be accessed efficiently from the apex by following the parent - children path. The construction of the pyramid proceeds bottom - up while the reconstruction from the apex is a top - down process. In such way we can reconstruct the whole pyramid by only locating the apex point.

6 A Sequence of 'Steps'

Now when analyzing a video sequence it is not necessary to compute one pyramid for each frame, it is enough to apply all the transformations to the apex and only to reconstruct the whole structure at the end of the process. Or we can rely to a few distinctive interest points (as done by several other approaches) the position and characteristics of which are known within the new object pyramid together with their mutual spatial relationships and track them while enforcing the preservation of the spatial relations like in the spring-system approach.

In that way, the complexity and the computation time are reduced what allows to adapt to the changes in the image frame in a more efficient way and being fast enough to deal with real time processing requirements.

6.1 Lossless Rotation

Another significant advantage of the above object pyramid is that the connectivity of both the foreground and the background is always preserved. This is not always true for other image processing tools (e.g. Photoshop), for example, when working with thin and elongated objects. Fig. 6 shows an example of a thin line (Fig. 6 a)) which is rotated by 50 degrees. As the new coordinates of the points of the line do not correspond to integer coordinates most of the image processing tools interpolate and resample the rotated coordinates in order to obtain the new position of the points. This results in a disconnected line (Fig. 6 b)) or in a thicker line if bilinear interpolation or anti-aliasing is applied. In the images of Fig. 6 b), the (red) stars mark the new rounded coordinates of each point and the black squares show the position of the points estimated by the processing tool. In our approach each pyramid level is a graph and the relations between adjacent regions are defined by the edges of the graph. When the top-down reconstruction is done, the position of each cell in each level is updated according to its correction vector but the edges of the graphs are always connecting the same vertices independently of their position. Therefore the region adjacency relations and connectivity are preserved (Fig. 6 c)).

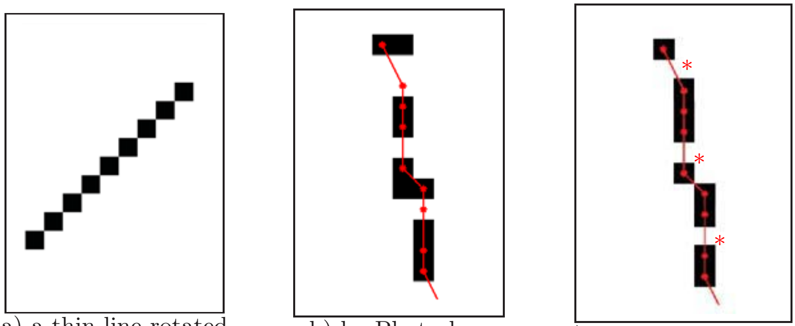
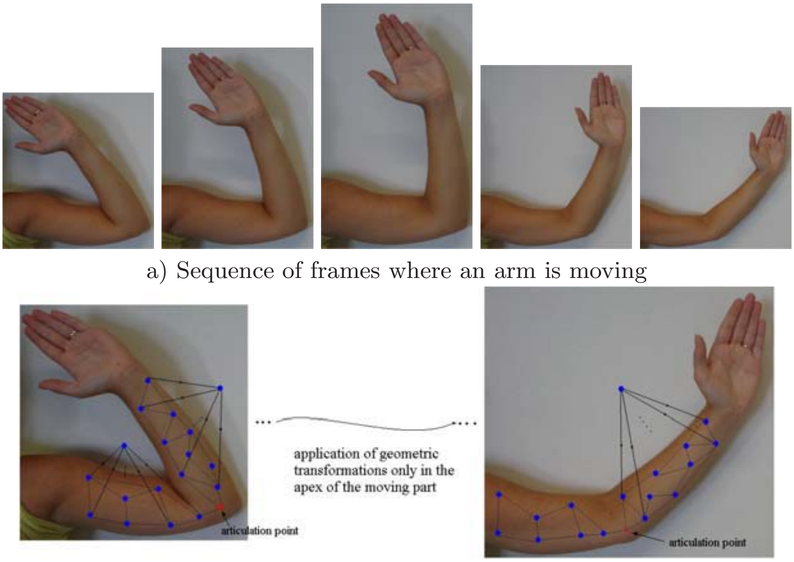

a) a thin line rotated   b) by Photoshop   c) as attributed graph

Fig. 6. 50^o rotation of a thin line

6.2 Articulated Objects

Our approach can be extended to articulated objects. Articulated objects are characterized by two or more rigid parts joint by articulation points. The nodes of the graph corresponding to the articulation point as well as the ones corresponding to the rigid parts are identified in the graph pyramid [47]. The nodes that compose each of the rigid parts will be merged in one apex in a certain pyramid level so that to follow the movement of each rigid part it is only needed to apply the geometric transformations in its apex and then doing the top-down reconstruction.

The top row of the Fig. 7 shows the movement of an arm in a sequence of 5 video frames. For the process of tracking the movement of the arm, in the first frame the structure is initialized and the pyramid is built (Fig. 7 a)). In this case only the lower part of the arm is moving, so that it will be only needed to apply all the transformations in the apex of the set of nodes that correspond to this part of the arm and reconstruct the graph at the end (7 b)). All the other nodes in the structure will remain in the same position. In that way, the tracking of articulated objects can be facilitated.


application of geometric transformations only in the apex of the moving part
articulation point
articulation point

a) Sequence of frames where an arm is moving

b) Structure initialization

c) Top-down reconstruction.

Fig. 7. Example with an articulated object

6.3 Adaptive Zoom–In: An Approaching Object

One application of the new object pyramid is the incremental update of the object description when the object approaches the camera. If the object gets closer to the camera, the distance camera-object decreases while the resolution of the object increases. We obtain a bigger picture of the object with more details to be inserted into the existing model of the object.

From the irregular graph pyramid perspective, this new image can be seen as a projection of the original base graph which includes more details. The pyramid will expand one level below the current base graph, this new base graph encodes both structures due to a higher resolution of the object. The new base graph is relinked to the current pyramid by finding the vertical connection between the new base graph and the existing upper part of the pyramid.

Assumption. We assume the approaching speed is not too fast. The current size of the target object cannot exceed twice as the one in previous image frame. This means the maximum scaling factor cannot exceed 2. Otherwise there might be a gap between the new base graph and the previous base graph so that we have to insert extra levels to bridge the new base graph with the old base graph.

The integration of several resolutions is also needed in surveillance applications involving multiple cameras observing the same area. The observed object will be closer to some cameras but further away from others. This creates the need to integrate the different views into a consistent description.

7 Conclusion

This paper presented some aspects of the development of image pyramids from stacks of arrays to a stack of graphs describing object's surfaces at multiple resolutions and multiple levels of abstraction. By decoupling the object's description from the projected view in an image frame into an object centered pyramid representation several operations become feasible: moving the object modifies only one cell of the structure much like the step of the foot when walking: the apex.

For a rigid object, the structure of the object pyramid is invariant to basic geometric transformation, such as translation, scaling and rotation. All the information about position and orientation of the object is concentrated in the apex. All the lower levels of a rigid object are invariant to translation, rotation and scale changes but still allowing accurate reconstruction of the object's geometry.

Tracking of structured objects is facilitated by the fact that the pyramid relates the different tracked points and can compensate tracking failures in case of partial occlusions. Articulation between rigid parts can be expressed by first selecting one of the related parts as the parent and describing the articulation by the change in Euler angles in the apex of the child. Different moving object pyramids can be related by superimposing a graph describing their spatial arrangement. In this graph the apexes of the objects appear as nodes related by edges describing the particular neighborhood relations. In some cases this graph could be embedded in $\mathbb{R}^3$ using a 3D combinatorial pyramid [49]. In the future several applications of the new concept are promising besides the spatio-temporal tracking of moving objects, i.e. the integration of views from different view points for surveillance.

Acknowledgements

The presented work is the result of the collaboration within the working group on robust hierarchies of PRIP. Substantial contributions came from Nicole Artner, Adrian Ion, Esther Antunez-Ortiz, Luis Alfredo Mateos, Dan Shao, Yll Haxhimusa, Helena Molina Abril, and Mabel Iglesias Ham. All these contributions are gratefully acknowledged.

References

1. Feldman, J.A., Ballard, D.H.: Connectionist models and their properties. Cognitive Science (6), 205–254 (1982)
2. Tsotsos, J.K.: On the relative complexity of passive vs active visual search. Intl. J. Computer Vision 7(2), 127–141 (1992)
3. Tsotsos, J.K.: How does human vision beat the computational complexity of visual perception? In: Pylyshyn, Z. (ed.) Computational Processes in Human Vision: An Interdisciplinary Perspective, pp. 286–338. Ablex Press, Notwood (1988)
4. Zeki, S.: A Vision of the Brain. Blackwell, Oxford (1993)
5. Pizlo, Z., Salach-Golyska, M., Rosenfeld, A.: Curve detection in a noisy image. Vision Research 37(9), 1217–1241 (1997)
6. Pizlo, Z.: Perception viewed as an inverse problem. Vision Research 41(24), 3145–3161 (2001)
7. Privitera, C.M., Stark, L.W.: Algorithms for defining visual regions-of-interest: Comparison with eye fixations. IEEE Tr. Pattern Analysis and Machine Intelligence 22(9), 970–982 (2000)
8. Jolion, J.M., Rosenfeld, A.: A Pyramid Framework for Early Vision. Kluwer Academic Publishers, Dordrecht (1994)
9. Tanimoto, S.L.: Paradigms for pyramid machine algorithms. In: Cantoni, V., Levialdi, S. (eds.) Pyramidal Systems for Image Processing and Computer Vision. NATO ASI Series, vol. F25, pp. 173–194. Springer, Heidelberg (1986)
10. Tanimoto, S.L., Klinger, A. (eds.): Structured Computer Vision: Machine Perception through Hierarchical Computation Structures. Academic Press, New York (1980)
11. Rosenfeld, A. (ed.): Multiresolution Image Processing and Analysis. Springer, Berlin (1984)
12. Tanimoto, S.L.: From pixels to predicates in pyramid machines. In: Simon, J.C. (ed.) Proceedings of the COST-13 workshop 'From the Pixels to the Features'. AFCET, Bonas, France (August 1988)
13. Bister, M., Cornelis, J., Rosenfeld, A.: A critical view of pyramid segmentation algorithms. Pattern Recognition Letters 11(9), 605–617 (1990)
14. Jolion, J.M., Montanvert, A.: The adaptive pyramid, a framework for 2D image analysis. Computer Vision, Graphics, and Image Processing: Image Understanding 55(3), 339–348 (1992)
15. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications COM-31(4), 532–540 (1983)
16. Anderson, C.H., Burt, P.J., van der Wal, G.S.: Change detection and tracking using pyramid transform techniques. In: Intelligent Robots and Computer Vision, September 16-20. SPIE, vol. 579, pp. 72–78 (1985)
17. Kropatsch, W.G., Haxhimusa, Y., Ion, A.: Multiresolution Image Segmentations in Graph Pyramids. In: Kandel, A., Bunke, H., Last, M. (eds.) Applied Graph Theory in Computer Vision and Pattern Recognition, vol. 52, pp. 3–41. Springer, New York (2007)

18. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recognition 26(3), 1277–1294 (1993)
19. Nacken, P.F.: Image segmentation by connectivity preserving relinking in hierarchical graph structures. Pattern Recognition 28(6), 907–920 (1995)
20. Keselman, Y., Dickinson, S.: Generic Model Abstraction from Examples. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 27(7), 1141–1156 (2005)
21. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man and Cybernetics 34, 334–352 (2004)
22. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4), Article 13 (2006)
23. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104, 90–126 (2006)
24. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding 81, 231–268 (2001)
25. Veenman, C.J., Reinders, M.J.T., Backer, E.: Resolving motion correspondence for densely moving points. IEEE Tr. Pattern Analysis and Machine Intelligence 23(1), 54–72 (2001)
26. Serby, D., Koller-Meier, S., Gool, L.V.: Probabilistic object tracking using multiple features. In: International Conference of Pattern Recognition, pp. 184–187. IEEE Computer Society, Los Alamitos (2004)
27. Shafique, K., Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. In: ICCV, Nice, France, vol. 1, pp. 110–115. IEEE Computer Society, Los Alamitos (2003)
28. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Tr. Pattern Analysis and Machine Intelligence 25(5), 564–575 (2003)
29. Csaba Beleznai, B.F., Bischof, H.: Human detection in groups using a fast mean shift procedure. In: International Conference on Image Processing, pp. 349–352. IEEE Computer Society, Los Alamitos (2004)
30. Yilmaz, A., Xin, L., Shah, M.: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Tr. Pattern Analysis and Machine Intelligence 26(11), 1531–1536 (2004)
31. Yunqiang, C., Yong, R., Huang, T.S.: JPDAF based HMM for real-time contour tracking. In: CVPR, Kauai, HI, USA, vol. 1, pp. 543–550. IEEE Computer Society, Los Alamitos (2001)
32. Felzenszwalb, P.F.: Pictorial structures for object recognition. International Journal for Computer Vision 61, 55–79 (2005)
33. Navaratnam, R., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Hierarchical part-based human body pose estimation. In: British Machine Vision Conference (2005)
34. Bray, M., Koller-Meier, E., Schraudolph, N.N., Gool, L.J.V.: Fast stochastic optimization for articulated structure tracking. Image Vision Comput. 25(3), 352–364 (2007)
35. Bobick, A.F., Wilson, A.D.: A state-based approach to the representation and recognition of gesture. IEEE Tr. Pattern Analysis and Machine Intelligence 19, 1325–1337 (1997)
36. Wilson, A.D., Bobick, A.F., Cassell, J.: Temporal classification of natural gesture and application to video coding. In: Conference on Computer Vision and Pattern Recognition, pp. 948–954 (1997)

37. Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. IEEE Tr. Pattern Analysis and Machine Intelligence 20(12), 1371–1375 (1998)
38. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 831–843 (2000)
39. Wren, C.R., Pentland, A.P.: Dynamic modeling of human motion. In: International Conference on Automatic Face and Gesture Recognition, pp. 14–16. IEEE Computer Society, Los Alamitos (1998)
40. Pavlovic, V., Rehg, J.M., Cham, T.J., Murphy, K.P.: A dynamic bayesian network approach to figure tracking using learned dynamic models. In: International Conference on Computer Vision, vol. 1, pp. 94–101. IEEE Computer Society, Los Alamitos (1999)
41. Kropatsch, W.G.: Building irregular pyramids by dual graph contraction. Vision, Image and Signal Processing 142(6), 366–374 (1995)
42. Kropatsch, W.G., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision Pyramids that do not Grow too High. Pattern Recognition Letters 26(3), 319–337 (2005)
43. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
44. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 731–737. IEEE, Los Alamitos (1997)
45. Haxhimusa, Y., Kropatsch, W.G., Pizlo, Z., Ion, A.: Approximative Graph Pyramid Solution of the E-TSP. Image and Vision Computing 27(7), 887–896 (2009)
46. Artner, N.M., López Mármol, S.B., Beleznai, C., Kropatsch, W.G.: Kernel-based Tracking Using Spatial Structure. In: Kuipers, A., Bettina Heise, L.M. (eds.) Challenges in the Biosciences: Image Analysis and Pattern Recognition Aspects. Proceedings of 32nd OEAGM Workshop, Linz, Austria, OCG-Schriftenreihe books@ocg.at, Österreichische Computer Gesellschaft, pp. 103–114, Band 232 (2008)
47. Artner, N.M., Ion, A., Kropatsch, W.G.: Rigid Part Decomposition in a Graph Pyramid. In: Eduardo Bayro-Corrochano, J.O.E. (ed.) The 14th International Congress on Pattern Recognition, CIARP 2009. LNCS. Springer, Heidelberg (2009)
48. Zhang, R., Tsai, P.S., Cryer, J., Shah, M.: Shape from Shading; A Survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(8), 609–706 (1999)
49. Illetschko, T.: Minimal combinatorial maps for analyzing 3d data. In: Technical Report PRIP-TR-110, Institute f. Automation 183/2, Dept. for Pattern Recognition and Image Processing, TU Wien, Austria (November 2006)

IX Feature Extraction, Clustering and Classification

A Simple Method for Eccentric Event Espial Using Mahalanobis Metric

Md. Haidar Sharif and Chabane Djeraba

University of Sciences and Technologies of Lille (USTL), France
{md-haidar.sharif,chabane.djeraba}@lifl.fr

Abstract. The paper presents an approach, which detects eccentric events in real time surveillance video systems (e.g., escalators), based on optical flow analysis of multitude behavour followed by *Mahalanobis* and χ^2 metrics. The video frames are flagged as normal or eccentric established on the statistical classification of the distribution of Mahalanobis distances of the normalized spatiotemporal information of optical flow vectors. Those optical flow vectors are computed from the small blocks of the explicit region of successive frames namely *Region of Interest Image* (RII), which is discovered by *RII Map* (RIIM). The RIIM is obtained from specific treatment of foreground segmentation of moving subjects. The method essentially has been tested against a single camera data-set.

1 Introduction

Video surveillance is commonly used in security systems, but requires more intelligent and more robust technical approaches. An automatic video surveillance is attractive because it pledges to replace the more costly option of staffing video surveillance monitors with human observers. There are many applications for systems that can detect emergencies and provide profitable and informative surveillance. For instance, escalators have become an importance portion of metropolitan life. The USCPSC estimates that there are approximately 7300 escalator-related injuries in the United States each year [1]. The USCPSC estimated an average of 5900 hospital emergency-room-treated injuries associated with escalators each year between 1990 and 1994. However, large-scale video surveillance of escalators would benefit from a system capable of recognizing eccentric (abnormal) events to make the system operators alert and fully informed. An *event* is said to be an observable action or change of state in a video stream that would be important for security management. For detecting events, authors in [2] focused on differences in the direction of motion and speed of persons, authors in [3] used optical flow features and support vector machine to detect surveillance events, while authors in [4] heavily relied on the optical flow concept to track feature points for each frame of a video. Optical flow features with Hidden Markov Models were used to detect emergency or eccentric events in the crowd [5,6] but those methods were not experimented on the real world video data-set. We will put forward an approach, which is based on statistical treatments of spatiotemporal (optical flow) patterns of human behaviours, to

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 417–424, 2009.

detect eccentric events essentially in unidirectional crowd flow (e.g., escalators). We get-go by calculating a Region of Interest Image Map (RIIM) during a period of time to extract the main regions of motion activity. The use of RIIM improves the quality of the results and reduces processing time which is an important factor for real-time applications. The optical flow information, calculated from the Region of Interest Images defined by RIIM in successive frames, of video reflects the crowd multi-modal behaviors as optical flow patterns variate in time. There is sufficient perturbation in the optical flow pattern in the crowd in case of abnormal and/or emergencies situations [see Fig. 1 (c) and (d)]. We calculate Mahalanobis distances using the extracted spatiotemporal information. Mahalanobis metric uses an appropriate correlation matrix to take into account of differences in variable variances and correlations between variables. We study the distribution of Mahalanobis distances along with a defined cutoff value T_d to make difference between normal and abnormal frames. To analyze the optical flow patterns of human crowds scenes, we have concentrated on escalator videos to use in our applications. One practical application of our approach is in the detection of real-time collapsing events, which could lead to perilous and inconsistent conditions. The exercised videos are from camera installed at an airport to monitor the situation of mainly escalator exits. The abstraction of the application is to have essentially escalator exits continuously observed to react quickly in the event of any collapsing. With this aim, cameras are installed in front of the exit locations to observe and send the video signal to a control room, where dedicated employees can monitor and respond to the collapsing situations.

The rest of this paper has been organized as follows: Section 2 delineates the proposed framework; Section 3 reports the experimental results; finally, Section 4 presents the conclusion of the work with few inklings for further investigation.

2 Proposed Approach

2.1 Region of Interest Image Map (RIIM)

The RIIM can be defined automatically by building a color histogram [see Fig. 1 (a) & (b) for escalator case], which is built from the accumulation of binary blobs of moving subjects, which were extracted following foreground segmentation method [7]. The adaptive background subtraction algorithm proposed by [7] is

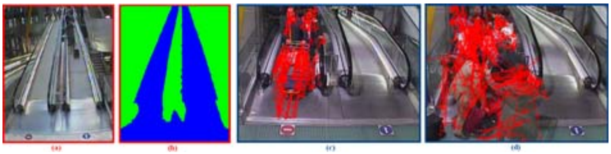


Fig. 1. (a) Camera view. (b) Generated *Region of Interest Image Map (RIIM)* and blue region on the RIIM recommends *Region of Interest Image* (RII). Ordered & disordered optical flow vectors in (c) & (d) limn *normal* and *abnormal* circumstances respectively.

able to model a background from a long training sequence with limited memory, works well on moving backgrounds, illumination changes, and compressed videos having irregular intensity distributions. The RIIM will be brought into existence mainly off-line. On-line is possible but it makes the system complicated. Off-line is better as the generated RIIM will be more significant and accurate when the video duration will be very long. RIIM improves the quality of the results and reduces processing time which is an imperative factor for real-time applications.

2.2 Spatiotemporal Information (ST-Info) Extraction

The Region of Interest Image (RII), ascertained by RIIM, is separated into small blocks. Once we define n (say 1500) points of interest in the RII, we track those points over the small blocks of two successive region of interest images using the combination feature tracker of Kanade-Lucas-Shi-Tomasi [8,9] easily. To get an acceptable distribution of optical flow pattern over the RII, we take into account vertical coordinate of each block. Consequently, a weighing coefficient λ is calculated according to the vertical coordinate of the block. A block far away from the camera has small vertical coordinate, as a result its λ should be large. Equally, block with large vertical coordinate get smaller λ. The value of λ heavily depends on the context of application and implementation. For our escalator videos data-set typically λ limits $0.6 \leq \lambda \leq 1$. Adjacent to camera (starting of the RII) region the value of $\lambda = 0.6$ suits well, whereas λ bears the maximum value 1 at the opposite end. We also take down the static and noise features. Static features are the features which moves less than two pixels. Noise features are the isolated features which have a big angle and distance difference with their near neighbors due to tracking calculation errors. Finally, for each frame [such as Fig. 1 (c) & (d)] irrespective of normal or eccentric events, we obtain an acceptable and workable spatiotemporal information, i.e., 5 features are observed in time and put in the form of a $n \times 5$ matrix $\mathbf{M(j)(k)}$ by dint of:

$$\mathbf{M(j)(k)} = \begin{bmatrix} x(1)(1) & x(2)(1) & x(3)(1) & x(4)(1) & x(5)(1) \\ . & . & . & . & . \\ x(1)(i) & x(2)(i) & x(3)(i) & x(4)(i) & x(5)(i) \\ . & . & . & . & . \\ x(1)(n) & x(2)(n) & x(3)(n) & x(4)(n) & x(5)(n) \end{bmatrix} \tag{1}$$

where $j = 1, 2, 3, 4, 5$; $k = 1, 2, \ldots, n$; i be a feature element in k; and $x(1)(i) \mapsto$ x-coordinate of the i, $x(2)(i) \mapsto y$-coordinate of the i, $x(3)(i) \mapsto x$-velocity with multiply by a weighing coefficient λ_i of the i, $x(4)(i) \mapsto y$-velocity with multiply by a weighing coefficient λ_i of the i, $x(5)(i) \mapsto$ acting motion direction of the i.

2.3 Statistical Treatments of the ST-Info

Normalization of Raw Data: A normalized value is a value that has been processed in a way that makes it possible to be efficiently compared against other values. For each column of $\mathbf{M(j)(k)}$, we calculate the *average* $\overline{x_j}$ and *standard deviation* σ_j. Subtracting the average $\overline{x_j}$ from each value in the columns of $x(j)(k)$, and then dividing by the standard deviation σ_j for that column in

$x(j)(k)$ generated a new matrix $z(j)(k)$ as: $\overline{x_j} = \frac{1}{n}\sum_{k=1}^{n} x(j)(k)$, $\sigma_j = \sqrt{\frac{\sum(x(j)(k)-\overline{x_j})^2}{n-1}}$, $z(j)(k) = \frac{x(j)(k)-\overline{x_j}}{\sigma_j}$. All values in $z(j)(k)$ are *dimensionless* and *normalized*, consequently, the new pattern of the $\mathbf{M(j)(k)}$ gives right of way:

$$\mathbf{Z(j)(k)} = \begin{bmatrix} z(1)(1) & z(2)(1) & z(3)(1) & z(4)(1) & z(5)(1) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z(1)(i) & z(2)(i) & z(3)(i) & z(4)(i) & z(5)(i) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z(1)(n) & z(2)(n) & z(3)(n) & z(4)(n) & z(5)(n) \end{bmatrix}. \tag{2}$$

A covariance matrix is merely collection of several variance-covariances in the form of a square matrix. But one problem with covariance is that it is sensitive to the scales. To obtain a more direct indication of how two components covary, we scale covariance to obtain correlation. Correlation is dimensionless while covariation is in units obtained by multiplying the units of each variable. Using $\mathbf{Z(j)(k)}$, scaling is performed by means of the following equations: $r_{pq} = \frac{S_{pq}}{S_p S_q}$, $S_{pq} = \frac{1}{n-1}\sum_{k=1}^{n}[z_p(k)z_q(k)]$, $S_l = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}[z_l(k)^2]}$, $\{p,q\} \in j$, $l \in \{p,q\}$.

Calculation of Mahalanobis Distance $D_m(i)$: In statistics, Mahalanobis distance is based on correlations between variables by which different patterns can be identified and analyzed. It is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e., not dependent on the scale of measurements. The region of constant Mahalanobis distance around the mean forms an ellipse in two dimensional space (i.e., when only 2 variables are measured), or an ellipsoid or hyperellipsoid when more variables are used. The Mahalanobis distance is the same as the Euclidean distance if the correlation matrix is the identity matrix. We calculate the Mahalanobis distance $D_m(i)$ for each row of the normalized matrix $\mathbf{Z(j)(k)}$ by multiplying the row by the *inverted correlation matrix*, then multiplying the resulting vector by the transpose of the row of the $\mathbf{Z(j)(k)}$, then dividing the obtained result by the degree of freedom, finally grasping square root of the up-to-the-minute result as:

$$D_m(i) = \sqrt{\left[\frac{z(1)(i)\ z(2)(i)\ z(3)(i)\ z(4)(i)\ z(5)(i)}{5}\right] \begin{bmatrix} 1 & r_{12} & r_{13} & r_{14} & r_{15} \\ r_{21} & 1 & r_{23} & r_{24} & r_{25} \\ r_{31} & r_{32} & 1 & r_{34} & r_{35} \\ r_{41} & r_{42} & r_{43} & 1 & r_{45} \\ r_{51} & r_{52} & r_{53} & r_{54} & 1 \end{bmatrix}^{-1} \begin{bmatrix} z(1)(i) \\ z(2)(i) \\ z(3)(i) \\ z(4)(i) \\ z(5)(i) \end{bmatrix}} \tag{3}$$

where the number of columns contained in $\mathbf{Z(j)(k)}$ is referred to as the degree of freedom which is 5 in this case. Geometrically, samples with an equal $D_m(i)$ lie on an ellipsoid (Mahalanobis Space). The $D_m(i)$ is small for samples lying on or close to the principal axis of the ellipsoid. Samples further away from the principal axis have a much higher $D_m(i)$. The larger the $D_m(i)$ for a sample is, the more likely the sample is an outlier. An outlier (extreme sample) is a sample that is very different from the average sample in the data set. An outlier may be an ordinary sample, but of which at least one attribute has been severely corrupted by a mistake or error (e.g., tracking calculation errors). An outlier may also be a bona fide sample, that simply turns out to be exceptional. Since Mahalanobis distance satisfies the conditions (symmetry, positivity, triangle inequality) of metric, it is a metric. The use of the Mahalanobis metric removes several limitations of the Euclidean metric: (i) it automatically accounts for the scaling of the coordinate axes, (ii) it corrects for correlation between the different features, (iii) it can provide curved as well as linear decision boundaries. But, there is a disbursement to be paid for those advantages. The computation of the correlation matrix can give rise to problems. When the investigated data are measured over a large number of variables, they can keep under control much redundant or correlated information. This is so-called *multicollinearity* in the data which leads to a singular correlation matrix that cannot be inverted. Another precinct for the calculation of the correlation matrix is that the number of samples in the data set has to be larger than the number of variables. Yet, in the proposed approach, both problems have been minimized by dint of 5 variables and tracking about 1500 samples (points of interest) in each frame respectively.

2.4 Classification of Mahalanobis Distances and T_d Estimation

Mahalanobis squared distances are calculated in units of standard deviation from the group mean. Therefore, the calculated circumscribing ellipse formed around the samples actually defines the one standard deviation of that group. This allows the designing of a statistical probability to that measurement. In theory, Mahalanobis squared distance is distributed as a χ^2 statistic with degree of freedom equal to the number of independent variables in the analysis. The χ^2 distribution has only one parameter called the degree of freedom. The shape of a χ^2 distribution curve is skewed for very small degrees of freedom and it changes drastically as the degrees of freedom increase. Eventually, for large degrees of freedom, the χ^2 distribution curve looks like a normal distribution curve. Like all other continuous distribution curves, the total area under a χ^2 distribution curve is 1.0. The *three sigma rule*, or *68-95-99.7 rule*, or *empirical rule*, states that for a normal distribution, about 68%, 95%, 99.7% of the values lie within 1, 2, and 3 standard deviation of the mean respectively. Clearly, almost all values lie within 3 standard deviations of the mean. Consequently, samples that have a squared Mahalanobis distance larger than 3 have a probability less than 0.01. These samples can be classified as members of *non-member group*. Samples those have squared Mahalanobis distances less than 3 are then classified as members of *member group*. The determination of the threshold depends on the application and the type of samples. In the proposed approach, we settle each $D_m(i)$ goes either *member group* or *non-member group*. Sample with a higher $D_m(i)$ than $\sqrt{3}$ is treated as *non-member group*, otherwise *member group*. *Member group* contains absolutely the samples of a normal event, whereas *non-member group* contains essentially samples of eccentric events (including outliers). Figure 2 depicts, while Mahalanobis metric produces elliptical cluster where samples are well correlated, Euclidean metric produces circular subsets. The *non-member group* consists of samples s_1, s_2, s_3, s_4, s_5, s_6, s_7, and the outlier s_8, while the *member group* groups the rest samples. Presuming in any *non-member group*, having M

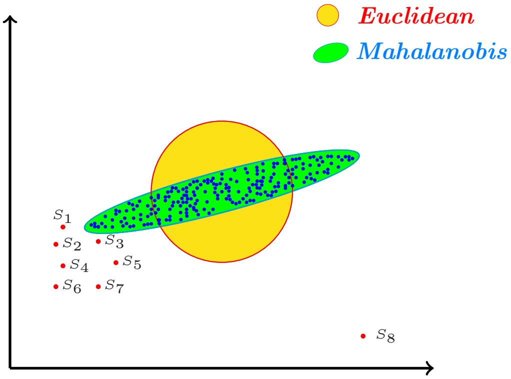


Fig. 2. Mahalanobis metric with respect to Euclidean metric. Samples s_1, s_2, s_3, s_4, s_5, s_6, s_7, and outlier s_8 go to *non-member group*, while the rests are *member group*.

samples including outliers (where also assuming that in general $M \gg outliers$ satisfies), we sum up their Mahalanobis distances as: $S_d = \sum_{i=1}^{M} D_m(i)$. Now, we transfer each S_d into a normalized distance (probability) value ranges between 0 and 1. The normalization may be done by using the simple formula like $1/log(S_d)$, but the normalized values fall into a congested range (scaling problem) which will arise problem specially in threshold selection. To solve the scaling problem, we take the advantage of cumulative distribution function (*cdf*), which has strict lower and upper bounds between 0 and 1, we can easily pick up the normalized distance of each S_d. Since all values of S_d are skewed to the right (positive-definite) and their variances are also large, we can use *Log-normal* distribution. Skewed distributions are particularly common when mean values are low, variances large, and values cannot be negative. Log-normal distributions are usually characterized in terms of the log-transformed variable, using as parameters the expected value, or mean (*location* parameter μ), of its distribution, and the standard deviation (*scale* parameter σ). The σ is entitled as *scale* as its value determines the *scale* or statistical dispersion of the probability distribution. If N_d be the normalized value of S_d, then N_d can be gently estimated by means of:

$$N_d = \frac{1}{2}\left[1 + erf\{\frac{log(S_d) - \mu}{\sigma\sqrt{2}}\}\right], \; erf(r) = \frac{2}{\sqrt{\pi}}\left[r - \frac{r^3}{3} + \frac{r^5}{10} - \frac{r^7}{42} + \ldots\right] \quad (4)$$

where erf is a $Gauss\ error\ function$ and $r = \frac{log(S_d)-\mu}{\sigma\sqrt{2}}$. Using Eq. 4, and placing congenial values of μ and σ (say $\mu = 0$, $\sigma = 5$) we can explicitly estimate the value of N_d between 0 and 1. Now, it is important to define an appropriate threshold T_d to make a distinction between normal and abnormal frames. We make a similitude measure between N_d and T_d to reach an explicit conclusion for each frame, i.e., a frame is said to be *eccentric if* $N_d > T_d$, otherwise *normal*. We estimate T_d from video stream which contains none but normal motions using:

$$T_d = \sqrt{\left[arg \max_{i=1\ldots f} [N_d]_i\right]^2 + \left[arg \min_{i=1\ldots f} \left[\frac{2}{\pi^2} \sum_{n=0}^{\infty} \frac{(-1)^n (N_d)^{2n+1}}{n!(2n+1)}\right]_i\right]^2} \quad (5)$$

where f be the total number of frames. The T_d extremely depends on the controlled environment. If the video stream changes, then T_d should be regenerated.

3 Experimental Results

To conduct experiments we used 16 real videos, taken in spanning days and seasons, of frame size 640×480 pixels, collected by cameras installed in an airport to monitor especially the escalator egresses, provided by a video surveillance company (escalator video frames in Fig. 3 have been cropped/grayed for confidential reasons). The videos were used to provide informative data for the security team who may need to take prompt actions in the event of a critical situation such as collapsing. Each video stream consists of normal and eccentric events. The normal situations correspond to crowd flows without any eccentric event on the escalator elsewhere. Eccentric events correspond to videos which contain collapsing events mostly in the escalator egresses. Generally, in the videos we have two escalators corresponding to two-way-traffic of opposite directions. The 1st image (from left) of Fig. 3 describes a scenario of a collapsing event in an escalator exit point. Some stuffs from a heavily loaded trolley have dropped just the egress point of the moving escalator which has caused one kind of emergency situation on the egress point. The 2nd image figures another example of an abnormal event where a wheel from the trolley has suddenly been broken off by the friction during its travel over an escalator and finally on the escalator exit point one kind of perilous and inconsistent circumstances has been come off. The situations have been detected by the proposed algorithm. But the algorithm does not work two of the video streams where video frames bear the situations like Fig. 3 (a) and (b) as the video sequences which include abnormal events have occurred with occlusion. Thus the quantity of extracted optical flow vectors is not sufficient to draw out abnormal frames. Of course, occlusion handling is a difficult part of optical flow technique. Occluded pixels violate a major assumption of optical

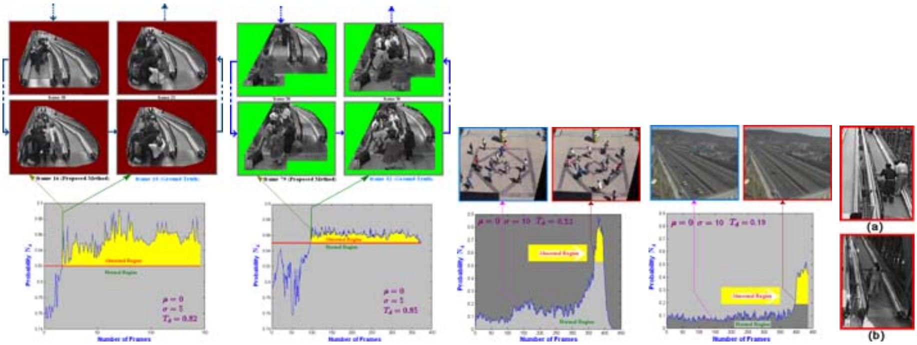


Fig. 3. Curves are the outputs of algorithm. Eccentric events on escalator exits (*from left* 1st two images), a sudden run of mob (3rd), turnabout of car on high-way (4th) have been detected. But occluded abnormal events in (a) and (b) can not be detected.

flow technique that each pixel goes somewhere. However, the detection results have been compared with ground truth. Ground truth is the process of manually marking what an algorithm is expected to output. Beyond the escalator unidirectional flow of mob videos, the method has been tested on the videos existing both normal and abnormal events, attributed 320×240 pixels, where the movements of people are random directions or where cars violate the traffic rules on the high-way, e.g., scenarios depicted on the 3rd and 4th images of Fig. 3. In the 3rd image, people has tended to leave their places with very quick motion. The 4th image concerns the scenario of breaking high-way traffic rules. When the car has tried to make a turnabout, it has broken the traffic rules which has been detected by the algorithm. The blue colored curves are the output of the algorithm. The yellow colored regions represent the abnormal motion activities.

4 Conclusion

We evinced a method which detects abnormal events in real time surveillance video systems. ST-Info has been extracted from the small blocks of the RII discovered by RIIM, which improves the quality of results and reduces processing time. The study of the distribution of Mahalanobis distances with predefined threshold T_d provides the knowledge of the state of abnormality. Efficacy of the algorithm has been evaluated on the real world crowd scenes. Obtained results, have been compared with ground truths, show the effectiveness of the method on detecting abnormalities. Yet, future work will make suit the method for the cases e.g., a normal event makes less visible or unclear an abnormal event, etc.

References

1. Commission, U.C.P.S.: Cpsc document #5111: Escalator safety. The United States Consumer Product Safety Commission (USCPSC), Washington, DC (2003)
2. Kawai, Y., Takahashi, M., Sano, M., Fujii, M.: High-level feature extraction and surveillance event detection. In: NHK STRL at TRECVID (2008)
3. Hao, S., Yoshizawa, Y., Yamasaki, K., Shinoda, K., Furui, S.: Tokyo Tech at TRECVID (2008)
4. Orhan, O.B., Hochreiter, J., Poock, J., Chen, Q., Chabra, A., Shah, M.: Content based copy detection and surveillance event detection. In: UCF at TRECVID (2008)
5. Andrade, E.L., Blunsden, S., Fisher, R.B.: Hidden markov models for optical flow analysis in crowds. In: ICPR 2006, pp. 460–463 (2006)
6. Andrade, E.L., Blunsden, S., Fisher, R.B.: Modelling crowd scenes for event detection. In: ICPR 2006, pp. 175–178 (2006)
7. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. Real-Time Imaging 11, 172–185 (2005)
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI 1981, pp. 674–679 (1981)
9. Shi, J., Tomasi, C.: Good features to track. In: CVPR 1994, pp. 593–600 (1994)

A Combine-Correct-Combine Scheme for Optimizing Dissimilarity-Based Classifiers[*]

Sang-Woon Kim[1] and Robert P.W. Duin[2]

[1] Dept. of Computer Science and Engineering, Myongji University, Yongin,
449-728 South Korea
kimsw@mju.ac.kr

[2] Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, The Netherlands
r.p.w.duin@tudelft.nl

Abstract. Recently, to increase the classification accuracy of dissimilarity-based classifications (DBCs), Kim and Duin [5] proposed a method of simultaneously employing fusion strategies in representing features (representation step) as well as in designing classifiers (generalization step). In this multiple fusion strategies, however, the resulting dissimilarity matrix is sometimes an *indefinite* one, causing problems in using the traditional pattern recognition tools after embedding the matrix in a vector space. To overcome this problem, we study a new way, named combine-correct-combine (CCC) scheme, of additionally employing an Euclidean correction procedure between the two steps. In CCC scheme, we first combine dissimilarity matrices obtained with different measures to a new dissimilarity representation using a representation combining strategy. Next, we correct the dissimilarity matrix using a pseudo-Euclidean embedding algorithm to improve the internal consistency of the matrix. After that, we again utilize the classifier combining strategies in the refined dissimilarity matrix to achieve an improved classification for a given data set. Our experimental results for well-known benchmark databases demonstrate that the CCC mechanism works well and achieves further improved results in terms of the classification accuracy compared with the previous multiple fusion approaches. The results especially demonstrate that the highest accuracies are obtained when the refined representation is classified with the trained combiners.

1 Introduction

In statistical pattern recognition, classification is performed in two steps: representation and generalization. In a case of dissimilarity-based classification (DBC) [9] [1], dissimilarity matrix is generated first from the training set in the representation step. Then, in the generalization step, classifiers are designed in the dissimilarity matrix.

[*] The work of the first author was done while visiting at Delft University of Technology, The Netherlands. We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-013-D00115).

[1] This methodology is not based on the feature measurements of the individual patterns, but rather on a suitable dissimilarity measure between them. An introduction to DBC will appear in a subsequent section.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 425–432, 2009.

On the other hand, combination systems which fuse "pieces" of information have received considerable attention because of its potential to improve the performance of individual systems [6], [7]. Recently, to increase the classification accuracy of DBCs, Kim and Duin [5] proposed a method of simultaneously employing multi-level fusion strategies in representing features (representation step) as well as in designing classifiers (generalization step). In [5], the authors first combined dissimilarity matrices obtained with different measures to a new representation matrix. Then, after training some base classifiers in the new representation matrix, they again combined the results of the base classifiers. In this multiple combining scheme, however, the representation matrix obtained is sometimes an *indefinite* one [8], causing problems in using the traditional pattern recognition tools after embedding the matrix in a vector space [10].

Combining dissimilarity matrices of metric measures sometimes leads to non-Euclidean ones, in which the metric requirement, such as the symmetry or the triangle inequality, is disobeyed. Duin and his colleagues [10] found that classifiers based on non-Euclidean dissimilarity representations may lead to better results than those based on transformed dissimilarity measures that are either Euclidean or have reduced non-Euclidean components [10]. Non-Euclidean vector spaces, however, are not well equipped with the tools for training classifiers; distances have to be computed in a specific way and are usually not invariant to orthogonal rotations. Also densities may not be properly defined, though some density-based classifiers can be used under some restrictions. So, Euclidean corrections called *Euclideanization* become of interest [8], [10].

To overcome the problem mentioned above, we study a new way, named combine-correct-combine (CCC) scheme, of additionally employing an Euclidean correction procedure between the two steps. In CCC scheme, we first combine dissimilarity matrices obtained with different measures to a new representation matrix using one of the representation combining strategies. We then correct the representation matrix using pseudo-Euclidean embedding algorithms to improve the internal consistency of the matrix. Finally, we again utilize the classifier combining strategies in the refined dissimilarity matrix to achieve an improved classification accuracy. Indeed, we show that by correcting the dissimilarity representation matrix resulted from combining various dissimilarity matrices, we can obtain a refined representation matrix, using which, in turn, the classifier combining strategies can be employed again to improve the classification accuracy. Our experimental results for benchmark databases demonstrate that the proposed CCC mechanism works well and achieves further improved accuracies compared with the previous multiple fusion approaches [5].

The main contribution of this paper is to demonstrate that combined DBCs can be optimized by employing an Euclidean correction procedure. This has been done by executing the correction procedure prior to the classifier combining process and by demonstrating its strength in terms of the classification accuracy. The reader should observe that this philosophy is *quite* simple and distinct from that used in [5].

2 Combine-Correct-Combine Scheme for DBCs

Foundations of DBCs: A dissimilarity representation of a set of samples, $T = \{x_i\}_{i=1}^{n} \in \Re^d$, is based on pairwise comparisons and is expressed, for example, as an $n \times m$

dissimilarity matrix $D_{T,Y}[i,j]$, where $Y = \{y_j\}_{j=1}^{m}$, a prototype set, is extracted from T, and the subscripts of D represent the set of elements on which the dissimilarities are evaluated. Thus, each entry $D_{T,Y}[i,j]$ corresponds to the dissimilarity between the pairs of objects $\langle x_i, y_j \rangle$, where $x_i \in T$ and $y_j \in Y$. Consequently, an object x_i is represented as a column vector as follows: $[d(x_i,y_1), d(x_i,y_2), \cdots, d(x_i,y_m)]^T$, $1 \leq i \leq n$. Here, the measure d is only required to be reflexive, i.e., $d(x,x) = 0$ for all x. Also, the dissimilarity matrix D is defined as a *dissimilarity space* on which the d-dimensional object, x, given in the feature space, is represented as an m-dimensional vector $\delta(x,Y)$, where if $x = x_i$, $\delta(x_i,Y)$ is the i-th row of D. In this paper, the dissimilarity matrix and its column vectors are simply denoted by $D(T,Y)$ and $\delta(x_i)$, respectively.

Representation Combining Strategies: Here, it is interesting to note that a number of distinct dissimilarity representations can be combined into a new one to obtain a more powerful representation in the discrimination. The idea of this *feature* combination is derived from the possibility that discriminative properties of different representations can be enhanced by a proper fusion [9] [2]. There are several schemes for combining multiple representations to solve a given classification problem. Some of them are *Average*, *Product*, *Min*, and *Max* rules. For example, in the *Average* rule, two dissimilarity matrices, $D^{(1)}(T,Y)$ and $D^{(2)}(T,Y)$, can be averaged to $(\alpha_1 D^{(1)}(T,Y) + \alpha_2 D^{(2)}(T,Y))$ after scaling with an appropriate weight, α_i, to guarantee that they all take values in a similar range. The details of these methods are omitted here, but can be found in [9].

Representation Correcting Strategies: A symmetric dissimilarity matrix $D \in \Re^{n \times n}$ can be embedded in a pseudo-Euclidean space $\Xi (= \Re^{(p,q)} = \Re^{(p)} \oplus \Re^{(q)})$, by an isometric mapping [3], [8], [9]. The pseudo-Euclidean space Ξ is determined by eigendecomposition of an Gram matrix, $G = -\frac{1}{2} J D^{*2} J$, derived from D, where J is the centering matrix [3] and D^{*2} is the square dissimilarity matrix. The details of the derivation can be found in the related literature including [1], [9]. In this decomposition, p positive and q negative eigenvalues arise, indicating the signature of Ξ. The axes of Ξ are constituted by $\sqrt{|\lambda_i|}\mu_i$, where λ_i and μ_i are the ith eigenvalue and the corresponding eigenvector of G, respectively.

There are several schemes for determining the pseudo-Euclidean space to refine the dissimilarity representation resulted from combining dissimilarity matrices. Some of them are briefly introduced as follows:

1. NON (non-refined space): This method is the same as the multiple fusion scheme in [5]. That is, the combiners are trained in non-refined matrix.

2. PES+ (pseudo Euclidean space): The most obvious correction for a pseudo-Euclidean space $\Xi = \Re^{(p,q)}$ is to neglect the negative definite subspace. This discarding results in a p-dimensional Euclidean space $\Re^{(p)}$ with many-to-one mappings to Ξ. Consequently, it is possible that the class overlap increases.

3. AES (associated Euclidean space): Since $\Re^{(p,q)}$ is a vector space, we can keep all dimensions when performing the isometric mapping, which implies that the vector coordinates are identical to those of Ξ, but we now use the norm and distance measure that are Euclidean.

[2] This is also related to a kind of clustering ensemble which combines similarity matrices [13].

[3] $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \in \Re^{n \times n}$, where I is the identity matrix and $\mathbf{1}$ is an n-elements vector of all ones. The details of the centering matrix can be found in [1].

4. AESR (associated Euclidean space reduction): In this correction, we can find an Euclidean subspace based on the $p'(\leq p)$ positive eigenvalues and $q'(\leq q)$ negative eigenvalues when computing a projection.

The details of the other correction procedures, such as DEC (dissimilarity enlargement by a constant), Relax (relaxation by a power transformation), and Laplace (Laplace transformation), are omitted here, but can be found in [3].

Classifier Combining Strategies: The basic strategy used in fusion is to solve the classification problem by designing a *set* of classifiers, and then combining the individual results obtained from these classifiers in some way to achieve reduced classification error rates. Therefore, the choice of an appropriate fusion method can further improve on the performance of the individual method. Various classifier fusion strategies have been proposed in the literature. Some of them are *Product*, *Sum*, *Average*, *Max*, *Min*, *Median*, *Majority vote*, and so on [6]. In addition, there are two commonly used approaches to implement multiple base-level classifiers; a fixed combiner and a trainable combiner. The fixed combiner has no extra parameter that need to be trained, while the trainable combiner needs additional training. For example, if a single training set is available, it is recommended to leave the base classifiers undertrained and subsequently complete the training of the combiner on the training set [7], [11]. Various classifier fusion strategies have been proposed in the literature - an excellent study is found in [7].

3 Combined Dissimilarity-Based Classifiers (CDBCs)

The reasons for combining several distinct dissimilarity representations and different dissimilarity-based classifiers will be investigated in the present paper. The proposed approach, which is referred to as a combined dissimilarity-based classifier (CDBC), is summarized in the following:

1. Select the input training data set T as a representative subset Y.
2. Compute dissimilarity matrices, $D^{(1)}(T,Y)$, $D^{(2)}(T,Y)$, $\cdots$, $D^{(k)}(T,Y)$, by using the k different dissimilarity measures for all $x \in T$ and $y \in Y$.
3. Combine the dissimilarity matrices, $\{D^{(i)}(T,Y)\}_{i=1}^{k}$, into new ones, $\{D^{(j)}(T,Y)\}_{j=1}^{l}$, by building an extended matrix or by computing their weighted average. Following this, correct the *new* matrices using an Euclidean correction procedure.
4. For any $D^{(j)}(T,Y)$, $(j = 1,\cdots,l)$, perform classification of the input, $\boldsymbol{z}$, with *combined* classifiers designed on the newly refined dissimilarity space as follows:
   (a) Compute a dissimilarity column vector, $\delta^{(j)}(\boldsymbol{z})$, for the input sample $\boldsymbol{z}$, with the same method as in measuring the $D^{(j)}(T,Y)$.
   (b) Classify $\delta^{(j)}(\boldsymbol{z})$ by invoking a group of DBCs as the *base* classifiers designed with n m-dimensional vectors in the dissimilarity space. The classification results are labeled as $class_1, class_2, \cdots$, respectively.
5. Obtain the final result from the $class_1$, $class_2$, $\cdots$, by combining the base classifiers designed in the above step, where the base classifiers are combined to form the final decision in the *fixed* or *trained* fashion.

The computational complexity of the proposed algorithm depends on the computational costs associated with the dissimilarity matrix. Thus, the time complexity and the space complexity of CDBC are $O(n^2 + d^3)$ and $O(n(n + d))$, respectively.

4 Experimental Results

Experimental Data: The proposed method has been tested by performing experiments on three benchmark databases, namely, the Yale[4], AT&T[5], and Nist38 databases.

The face database of Yale contains 165 gray scale images of 15 individuals. The size of each image is 243×320 pixels, for a total dimensionality of 77760 pixels. To reduce the computational complexity of this experiment, facial images of Yale database were down-sampled into 178×236 pixels and then represented by a centered vector of normalized intensity values. The face database of AT&T consists of ten different images of 40 distinct subjects, for a total of 400 images. The size of each image is 112×92 pixels, for a total dimensionality of 10304 pixels. The data set captioned Nist38 chosen from the NIST database [12] consists of two kinds of digits, 3 and 8, for a total of 1000 binary images. The size of each image is 32×32 pixels, for a total dimensionality of 1024 pixels.

Experimental Method: All our experiments were performed with a leave-one-out (LOO) strategy. To classify an image of an object, we removed the image from the training set and computed the dissimilarity matrix with the $n-1$ images. This process was repeated n times for every image, and a final result was obtained by averaging the results of each image. To compute the dissimilarity matrix, we first selected all training samples as the representative. We then measured the dissimilarities between them using four systems: Euclidean distance (ED), Hamming distance (HD)[6], the regional distance (RD)[7], and the spatially weighted gray-level Hausdorff distance (WD)[8] measures.

First of all, to investigate the representation combination, we experimented with three *Average* methods: Ex-1, Ex-2, and Ex-3. In Ex-1, two dissimilarity matrices obtained with ED and RD measures are averaged to a new representation after normalization, where the scaling factors are $\alpha_i = \frac{1}{2}, i = 1, 2$. In Ex-2, three dissimilarity matrices obtained with ED, RD, and HD measures are averaged to a new one with $\alpha_i = \frac{1}{3}, i = 1, 2, 3$. In Ex-3, four dissimilarity matrices measured with ED, RD, HD, and WD are averaged with $\alpha_i = \frac{1}{4}, i = 1, \cdots, 4$. In general, $\alpha_i = \frac{1}{N}$, where N is the number of matrices to be combined.

Next, to improve the internal consistency of the combined matrices, we refined them with three kinds of Euclidean corrections: AES, PES+, and AESR. In AES, all dimensions are kept when mapping the matrix onto a pseudo-Euclidean subspace. In PES+,

[4] http://www1.cs.columbia.edu/ belhumeur/pub/images/yalefaces

[5] http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html

[6] Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. For binary strings α and β, for example, the Hamming distance is equal to the number of ones in $\alpha \oplus \beta$ ($\oplus$ means *Exclusive*-OR operation).

[7] The regional distance is defined as the average of the minimum difference between the gray value of a pixel and the gray value of each pixel in a 5×5 neighborhood of the corresponding pixel. In this case, the regional distance compensates for a displacement of up to three pixels of the images.

[8] In WD, we compute the dissimilarity directly from input gray-level images without extracting the binary edge images from them. Also, instead of obtaining the distance on the basis of the entire image, we use a spatially weighted mask, which divides the image region into several subregions according to their importance.

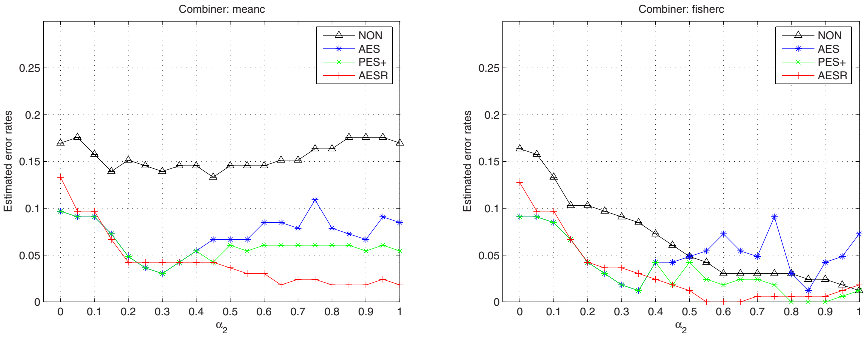


Fig. 1. A comparison of the error rates of two combiners, *meanc* and *fisherc*, obtained with Ex-1 method for Yale. Here $\alpha_2 (= 1 - \alpha_1)$ is the scaling factor of the dissimilarity matrix. Four markers describing the same Euclidean corrections are connected by lines to enhance the visibility.

a subspace is found based on the positive eigenvalues only, while, in AESR, the subspace is found such that at least a fraction of the negative eigenvalues is preserved. For instance, in this experiment, the cumulative fractions of the positive and negative eigenvalues were 0.9 and 0.1, respectively.

Finally, in CDBCs, l was set as 1; only one combination of dissimilarity matrices was made. Also, after training three base classifiers, all of the results were combined in *trainable* fashion. Here, the three base classifiers and the three trained combiners were implemented with PRTools [9], and named *nmc*, *ldc*, *knnc*, *meanc*, *fisherc*, and *naivebc*, respectively.

Experimental Results: First of all, to examine the rationality of employing the refining techniques in CDBCs, the classification error rates of two combiners, *meanc* and *fisherc*, were evaluated with Ex-1 method for Yale database. Here, combining dissimilarity matrices was done with 21 different scaling factors; $\alpha_2 = 0.0, 0.05, \cdots, 1.0$ and $\alpha_1 = 1 - \alpha_2$. Then, the resulted matrices were corrected in three ways: AES, PES+, and AESR. Finally, the results of the base classifiers were combined in trained fashion. Fig. 1 shows a comparison of the error rates obtained with Ex-1 method for Yale.

From the figure, it should be observed that the classification accuracies of the combiners trained in the refined matrix can be improved. This is clearly shown from the results of *meanc* (see the left picture), where the error rates of a NON-refined matrix is shown with the connected lines of $\triangle$ marker, and those of three refined matrices with $*$, $\times$, and $+$ markers. Similar, not necessarily the same, characteristics could also be observed in the results of *fisherc* (see the right picture). The details of the results are omitted here in the interest of compactness.

In order to further investigate the advantage gained with utilizing the CCC scheme, we repeated the experiments (of estimating error rates) in Ex-1, Ex-2, and Ex-3. Table 1 shows the estimated error rates of CDBCs designed as the base classifiers and the trainable combiners for the three experimental databases. In the table, the values underlined are the *lowest* ones in the 24 error rates (12 for the base classifiers and 12 for the trained combiners per each fusion method).

[9] PRTools is a Matlab toolbox for pattern recognition (refer to http://prtools.org/).

Table 1. A comparison of the estimated error rates of DBCs designed as the base classifiers and the trainable combiners for the three databases, where the values underlined are the *lowest* ones in the 24 error rates of the base and trained combiners per each combining method

experimental databases	combining method	correcting method	base classifiers			combiners		
			nmc	*ldc*	*knnc*	*meanc*	*fisherc*	*naivebc*
Yale	Ex-1	NON	0.2242	0.0485	0.1939	0.1455	0.0485	0.1455
		AES	0.1030	0.0485	0.1879	0.0667	0.0485	0.1091
		PES+	0.1091	0.0424	0.1879	0.0606	0.0424	0.1030
		AESR	0.1091	0.0121	0.1939	0.0364	0.0121	0.0970
	Ex-2	NON	0.2061	0.0909	0.1939	0.1455	0.0909	0.1697
		AES	0.1273	0.0182	0.1879	0.0424	0.0182	0.1091
		PES+	0.1273	0.0182	0.1879	0.0424	0.0182	0.1091
		AESR	0.1273	0.0303	0.1939	0.0424	0.0303	0.1152
	Ex-3	NON	0.2121	0.0485	0.2121	0.1576	0.0485	0.1818
		AES	0.1576	0.0848	0.2061	0.1273	0.0848	0.1697
		PES+	0.1576	0.0485	0.1879	0.0970	0.0485	0.1576
		AESR	0.1576	0.0121	0.2061	0.0485	0.0121	0.1515
AT&T	Ex-1	NON	0.2350	0.0075	0.0425	0.0300	0.0075	0.0300
		AES	0.0600	0.0250	0.0175	0.0275	0.0325	0.0300
		PES+	0.0575	0.0525	0.0150	0.0300	0.0500	0.0250
		AESR	0.0700	0.0150	0.0175	0.0175	0.0250	0.0225
	Ex-2	NON	0.1950	0.0050	0.0225	0.0175	0.0050	0.0175
		AES	0.0375	0.0350	0.0050	0.0125	0.0300	0.0100
		PES+	0.0375	0.0550	0.0050	0.0150	0.0475	0.0125
		AESR	0.0375	0.0050	0.0075	0.0050	0.0050	0.0075
	Ex-3	NON	0.2125	0.0050	0.0250	0.0175	0.0075	0.0175
		AES	0.0375	0.0675	0.0050	0.0150	0.0475	0.0125
		PES+	0.0375	0.0925	0.0075	0.0150	0.0700	0.0150
		AESR	0.0450	0.0025	0.0100	0.0025	0.0025	0.0100
Nist38	Ex-1	NON	0.1220	0.0220	0.0190	0.0190	0.0190	0.0220
		AES	0.0890	0.0670	0.0110	0.0370	0.0660	0.0220
		PES+	0.0930	0.0610	0.0130	0.0460	0.0130	0.0190
		AESR	0.0930	0.0260	0.0110	0.0250	0.0110	0.0110
	Ex-2	NON	0.1230	0.0210	0.0170	0.0170	0.0170	0.0220
		AES	0.0900	0.1060	0.0110	0.0490	0.1060	0.0210
		PES+	0.0920	0.0720	0.0150	0.0490	0.0150	0.0160
		AESR	0.0910	0.0290	0.0120	0.0240	0.0120	0.0120
	Ex-3	NON	0.1660	0.0180	0.0270	0.0180	0.0270	0.0270
		AES	0.0990	0.1990	0.0140	0.1010	0.1990	0.0350
		PES+	0.1010	0.1120	0.0140	0.0560	0.0140	0.0170
		AESR	0.0990	0.0240	0.0140	0.0180	0.0140	0.0130

The observations obtained from the table are the followings:

- The best Ex-3 results are usually better than the best Ex-2 and Ex-1 results, so combining dissimilarity matrices is helpful.

- The corrected results (AES, PES+ and AESR) are sometimes better than the original results (NON). So correction can be helpful, but sometimes it is not.

- The results *fisherc* as a trainable combining classifier are about equal to those of the best base classifier (usually *ldc*, sometimes *knnc*) and it thereby operates as a selector. Consequently, the combining classifier makes the system more robust.

Finally, we didn't present standard deviations in Table 1 to save some space as we don't claim that some improvements are significant. A more robust analysis can be performed in terms of quantitative measures such as the kappa or tau coefficients [2].

5 Conclusions

In this paper, we proposed to utilize the combine-correct-combine (CCC) scheme to optimize dissimilarity-based classifications (DBCs). The CCC scheme involves a step wherein the combined dissimilarity matrix is corrected prior to employing the classifier combining strategies to improve the internal consistency of the dissimilarity matrix. The presented experimental results for three benchmark databases demonstrate that the studied CCC mechanism works well and achieves robust, good results. Despite this success, problems remain to be addressed. First, classification performance could be improved furthermore by developing an optimal Euclidean correction and by designing suitable combiners in the refined dissimilarity space. Then, the experimental results also show that the highest accuracies are achieved when the refined representation is classified with the trained combiners. The problem of theoretically analyzing this observation remains unresolved. Future research will address these concerns.

References

1. Borg, I., Groenen, P.: Morden Mutlidimensional Scaling: Theory and Applications. Springer, New York (1997)
2. Congalton, R.G.: A review of assessing the accuracy of classifications of remotely sensed data. Remote Sensing of Enviroment 37, 35–46 (1991)
3. Duin, R.P.W., Pekalska, E., Harol, A., Lee, W.: On Euclidean corrections for non-Euclidean dissimilarities. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 551–561. Springer, Heidelberg (2008)
4. Haasdonk, H., Burkhardt, B.: Invariant kernels for pattern analysis and machine learning. Machine Learning 68, 35–61 (2007)
5. Kim, S.-W., Duin, R.P.W.: On optimizing dissimilarity-based classifier using multi-level fusion strategies. Journal of The Institute of Electronics Engineers of Korea 45-CI(5), 15–24 (2008) (in korean); A preliminary version of this paper was presented at the 20th Canadian Conference on Artificial Intelligence, Montreal, Canada. LNCS (LNAI), vol. 4509, pp. 110–121 (2007)
6. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. and Machine Intell. 20(3), 226–239 (1998)
7. Kuncheva, L.I.: Combining Pattern Classifiers - Methods and Algorithms. John Wiley & Sons, New Jersey (2004)
8. Munoz, A., de Diego, I.M.: From indefinite to positive semi-definite matrices. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 764–772. Springer, Heidelberg (2006)
9. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications. World Scientific Publishing, Singapore (2005)
10. Pekalska, E., Harol, A., Duin, R.P.W., Spillmann, B., Bunke, H.: Non-Euclidean or non-metric measures can be informative. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 871–880. Springer, Heidelberg (2006)
11. Todorovski, L., Dzeroski, S.: Combining classifiers with meta decision trees. Machine Learning 50(3), 223–249 (2003)
12. Wilson, C.L., Garris, M.D.: Handprinted Character Database 3, Technical report, National Institute of Standards and Technology, Gaithersburg, Maryland (1992)
13. Zhou, Z.-H., Tang, W.: Clusterer ensemble. Knowledge-Based System 19, 77–83 (2006)

BR: A New Method for Computing All Typical Testors

Alexsey Lias-Rodríguez[1] and Aurora Pons-Porrata[2]

[1] Computer Science Department
lias@csd.uo.edu.cu
[2] Center for Pattern Recognition and Data Mining
Universidad de Oriente, Santiago de Cuba, Cuba
aurora@cerpamid.co.cu

Abstract. Typical testors are very useful in Pattern Recognition, especially for Feature Selection problems. The complexity of computing all typical testors of a training matrix has an exponential growth with respect to the number of features. Several methods that speed up the calculation of the set of all typical testors have been developed, but nowadays, there are still problems where this set is impossible to find. With this aim, a new external scale algorithm *BR* is proposed. The experimental results demonstrate that this method clearly outperforms the two best algorithms reported in the literature.

Keywords: typical testors, feature selection.

1 Introduction

One of the problems in Pattern Recognition is Feature Selection, which consists on finding the features that provide relevant information in the classification process. In the Logical Combinatorial Pattern Recognition [1] feature selection is commonly carried out by using Testor Theory [2]. In this theory, a testor is defined as a set of features that distinguishes the objects of different classes. A testor is called irreducible (typical) if none of its proper subsets is a testor. When we refer to typical testors (TT), we restrict us to typical Zhuravlev's testors, where classes are crisp and disjoint sets, the comparison criteria for features are Boolean and the similarity measure assumes two objects as different if they are so in at least one of the features.

Typical testors have been widely used to evaluate the feature relevance [3] and as support sets in classification algorithms [4]. In Text Mining, they have also been used for text categorization [5] and document summarization [6]. Several algorithms have been developed to calculate the typical testors. They can be classified according to its computational strategy into two categories: external and internal scale algorithms. The first perform the TT calculation by generating elements of the power set of features in a predefined order, but trying to avoid the analysis of irrelevant subsets. The second ones explore the internal structure of the training matrix and find conditions that guarantee the testor property. In this paper, we focus on the first strategy.

The complexity of computing all typical testors has an exponential growth with respect to the number of features. Methodologies that speed up the calculation of typical testors have been developed, but nowadays, there are still problems where the set of

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 433–440, 2009.

all typical testors is impossible to find. Therefore, it is very important to develop better algorithms for obtaining typical testors. The external scale methods *LEX* [7] and *CT-EXT* [8] are reported to be the most efficient ones.

With this aim, we propose *BR*, a new external scale method that avoids the analysis of a greater number of irrelevant subsets and efficiently verifies the testor property by profiting from the computer bit operations. The method name is due to these Binary operations and its Recursive nature. The experimental results demonstrate that this method clearly outperforms the two best algorithms reported in the literature [8].

2 Basic Concepts

Before presenting our method, we review the main definitions of the Testor Theory and we define the basic concepts of this method.

Let *TM* be a training matrix containing *m* objects described in terms of *n* features $\Re=\{X_1,\ldots,X_n\}$ and distributed into *r* classes $\{C_1,\ldots,C_r\}$. Each feature X_i takes values in a set D_i, $i=1,\ldots,n$. A comparison criterion of dissimilarity $\psi_i : D_i \text{ x } D_i \rightarrow \{0,1\}$ is associated to each X_i (0=similar, 1=dissimilar). Applying these comparison criteria for all possible pairs of objects belonging to different classes in *TM*, a Boolean dissimilarity matrix, denoted by *DM*, is built. Notice that the number of rows in *DM* is $m' = \sum_{i=1}^{r-1}\sum_{j=i+1}^{r}\left|C_i\right|\left|C_j\right|$, where $|C_i|$ denotes the number of objects in the class C_i.

Let *p* and *q* be two rows of *DM*. *p* is a *subrow of q* if in all columns where *p* has 1, *q* has also it. A row *p* of *DM* is called *basic* if no row in *DM* is a subrow of *p*. The submatrix of *DM* containing all its basic rows (without repetitions) is called a *basic matrix* (*BM*). Then, a *testor* is a subset of features $\tau=\{X_{i_1},\ldots,X_{i_s}\}$ of *TM* for which a whole row of zeros does not appear in the remaining submatrix of *BM*, after eliminating all columns corresponding to the features in $\Re\backslash\tau$. τ is a typical testor if there is no proper subset of τ that meets the testor condition [2]. Commonly, algorithms used for computing typical testors make use of *BM* instead of *DM* due to the substantial reduction of rows.

Let $(a_1,\ldots,a_u)$ be a binary *u*-tuple of elements, $a_i\in\{0,1\}$, $i=1,\ldots,u$. We call *cardinal of a binary u-tuple* to the number of its elements (i.e., *u*). The column corresponding to a feature *X* in *BM* is a binary *u*-tuple, whose cardinal is the number of rows in *BM*. We will denote this *u*-tuple by c_X. We also define logical operations on binary *u*-tuples as follows:

$(a_1, a_2, \ldots, a_u) \vee (e_1, e_2, \ldots, e_u) = (a_1 \vee e_1, a_2 \vee e_2, \ldots, a_u \vee e_u)$

$(a_1, a_2, \ldots, a_u) \wedge (e_1, e_2, \ldots, e_u) = (a_1 \wedge e_1, a_2 \wedge e_2, \ldots, a_u \wedge e_u)$

$\neg(a_1, a_2, \ldots, a_u) = (\neg a_1, \neg a_2, \ldots, \neg a_u)$

$(a_1, a_2, \ldots, a_\mathrm{u}) \otimes (e_1, e_2, \ldots, e_u) = (a_1 \otimes e_1, a_2 \otimes e_2, \ldots, a_u \otimes e_u)$, where $\otimes$ denotes the XOR operation.

(1,…,1) and (0,…,0) represent binary *u*-tuples in which all elements are one and zero, respectively.

The notation $[X_1,\ldots,X_s]$, $X_i\in\Re$, is used to represent an ordered list of features and $last([X_1,\ldots,X_s])$ denotes the last element in the list, i.e. X_s. A list does not contain features is denoted as []. We call *length of a list l*, denoted as $|l|$, to the number of its features. All basic operations of the set theory (difference, intersection, subset or sublist, etc.) can be defined on ordered lists of features in a similar way. With the symbol + we denote the concatenation between ordered lists of features.

Let l be an ordered list of features. The notation $[l]$ represents a unitary list composed by the list l. Hereafter, by list we will understand an ordered list.

Definition 1. Let $l = [X_1,\ldots,X_s]$ be a feature list. We call *acceptance mask of l*, denoted as am_l, to the binary u-tuple in which the ith element is 1 if the ith row in *BM* has at least a 1 in the columns corresponding to the features of l and it is 0 otherwise.

Definition 2. Let $l = [X_1,\ldots,X_s]$ be a feature list. We call *compatibility mask of l*, denoted as cm_l, to the binary u-tuple in which the ith element is 1 if the ith row in *BM* has an only 1 in the columns corresponding to the features of l and it is 0 otherwise.

Notice that the cardinal of both am_l and cm_l is the number of rows in *BM*.

Example 1. Let $l_1 = [X_1, X_2]$, $l_2 = [X_5,X_6,X_7,X_8,X_9]$ and $l_3 = [X_1, X_2, X_8]$ be feature lists of a basic matrix *BM*. Its acceptance and compatibility masks are the following:

$$BM = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{array}{l} am_{l_1} = (1,1,0,0,1) \\ am_{l_2} = (1,1,1,1,1) \\ am_{l_3} = (1,1,0,1,1) \\ cm_{l_1} = (1,1,0,0,1) \\ cm_{l_2} = (1,1,0,0,0) \\ cm_{l_3} = (0,1,0,1,1) \end{array}$$

Proposition 1. A feature list $l = [X_1,\ldots,X_s]$ is a testor if and only if $am_l = (1,\ldots,1)$.

Definition 3. Let $l = [X_1,\ldots,X_s]$ be a feature list and $X \in \Re$. A row p in *BM* is a *typical row of X with respect to l* if it has a 1 in the column corresponding to X and zero in all the columns corresponding to the features in $l \setminus [X]$.

Notice that, by typical testor definition, a feature list l is a typical testor if l is a testor and satisfies the typicity property, i.e. for every feature $X \in l$ there is at least a typical row of X with respect to l.

Proposition 2. Let $l = [X_1,\ldots,X_s]$ be a feature list and $X \notin l$ a feature of *BM*. The acceptance mask of the list $l + [X]$ is calculated as follows: $am_{l+[X]} = am_l \vee c_X$.

Proposition 3. Let $l = [X_1,\ldots,X_s]$ be a feature list and $X \notin l$ a feature of *BM*. The compatibility mask of the list $l + [X]$ is calculated as follows:

$$cm_{l+[X]} = ((cm_l \otimes c_X) \wedge cm_l) \vee (\neg am_l \wedge c_X)$$

Notice that propositions 2 and 3 allow the updating of acceptance and compatibility masks, respectively when a new feature is added to a feature list.

Proposition 4. Let $l = [X_1,\ldots,X_s]$ be a feature list and $X \notin l$ a feature of *BM*. If at least one of the following conditions is satisfied:

1. $am_{l+[X]} = am_l$
2. $\exists X_i \in l$ such that $cm_{l+[X]} \wedge c_{X_i} = (0,\ldots,0)$

Then, X does not form a typical testor with l. In this case, we will say that X is *exclusive with l.*

The condition 1 means that X has no typical rows with respect to l and the second one indicates that X_i loses all its typical rows due to X.

Notice that X_6 is exclusive with l_1 in the Example 1, since X_2 holds that $cm_{l_1+[X_6]} \wedge c_{X_2} = (1,0,1,0,1) \wedge (0,1,0,0,0) = (0,0,0,0,0)$. Notice also that X_8 is non-exclusive with l_1, because $am_{l_1+[X_8]} \neq am_{l_1}$ ($l_3=l_1+[X_8]$), $cm_{l_3} \wedge c_{X_1} = (0,0,0,0,1)$ and $cm_{l_3} \wedge c_{X_2} = (0,1,0,0,0)$.

Proposition 5. Let $l = [X_1,\ldots,X_s]$ be a feature list and $X \notin l$ a feature of *BM*. $l + [X]$ is a typical testor if and only if X is non-exclusive with l and $am_{l+[X]} = (1,\ldots,1)$.

The first condition means that all features of l+[X] have at least a typical row with respect to l+[X]. The second one guarantees that l+[X] is a testor, by proposition 1.

Definition 4. Let $l = [X_1,\ldots,X_s]$ be a feature list, p an integer such that $1 \leq p \leq s+1$ and $X \notin l$ a feature of *BM*. We call *substitution of X in l according to p*, denoted as *subst*(l,X,p), to the list $l' = [X_1,\ldots,X_{p-1},X]$. If l = [] then *subst*(l,X,1) = [X].

Notice that if $p = s+1$, *subst*(l,X,p) is the list l+[X].

Definition 5. Let $l=[X_{i_1},\ldots,X_{i_p}]$ and $l'=[X_{j_1},\ldots,X_{j_q}]$ be feature lists such that $l \cap l'$=[]. We call *non-exclusive list of l with respect to l'*, denoted as *nonExcl*(l,l'), to the list composed by the features $X_{i_k} \in l$ such that X_{i_k} is non-exclusive with l' and $l' + [X_{i_k}]$ is not a typical testor.

For instance, in the basic matrix of Example 1, *nonExcl*(l_2, l_1) = [X_7,X_8]. Notice that X_5 and X_9 are non-exclusive with l_1, but [X_1, X_2, X_5] and [X_1, X_2, X_9] are typical testors.

Definition 6. Let $l=[X_{i_1},\ldots,X_{i_p}]$ and $l'=[X_{j_1},\ldots,X_{j_q}]$ be feature lists such that $l \cap l'$=[]. We call *typical list of l with respect to l'*, denoted as *TypL*(l,l') to the list composed by the lists $l' + [X_{i_k}]$ such that $X_{i_k} \in l$ and $l' + [X_{i_k}]$ is a typical testor.

For instance, in the basic matrix of Example 1, *TypL*(l_2,l_1) = [[X_1,X_2,X_5],[X_1,X_2,X_9]].

3 BR Method

The proposed method firstly rearranges the rows and columns of *BM* in order to reduce the search space of typical testors. The row with the minimum number of 1's and the maximum number of 1's in the columns of *BM* where it has a 1 is put as the first row (see Steps 1a and 1b). In the Example 1, the two first rows have two 1's, but the first row stays there, since it has four 1's in the columns where it has a 1 (X_1, X_8). The rearrangement of columns (see Step 1c) allows the algorithm finishes as soon as possible, i.e., when the feature to be analyzed has a zero in the first row of *BM*. Notice that all possible combinations of the remaining features will not be testors. The rearrangement of columns also attempts to reduce the likelihood of the features to be analyzed being non-exclusive with a feature list, and therefore, to minimize the length of the feature lists that must be examined.

The underlying idea of *BR* method is firstly to generate feature lists that satisfy the typicity property and secondly to verify the testor condition. Like *LEX* and *CT-EXT* algorithms, our method explores the power set of features starting from the first feature in *BM* and generates candidate feature lists to be typical testors. Once a candidate feature list has been generated, the typicity and testor properties are verified by using propositions 1, 4 and 5. Notice that these propositions are based on acceptance and compatibility masks.

Given a candidate feature list *L*, *BR* method builds the list *LP* composed by the features X_i that are non-exclusive and do not form a typical testor with *L*. It means that L+[X_i] needs more features to form a typical testor. Unlike previous algorithms, which attempt to find these features in *BM*, our method restricts the search to the features in *LP*. This fact is based on the following proposition:

Proposition 6. Let l=[X_1,…,X_s] be a feature list and $X \notin l$ a feature of *BM*. If *X* is exclusive with *l*, then it will also be exclusive with any list *l'*, such that $l \subseteq l'$ and $X \notin l'$.

Notice that the features X_i that form a typical testor with *L* are not included in *LP*. In this case, these typical testors are stored in *TTR*. Then, the first feature in *LP* is added to *L* and the remaining features in *LP* that are non-exclusive with *L* are selected again. This process is repeated until all typical testors containing the first feature in *BM* are found. Then, the algorithm starts from the second feature in *BM* and repeats all steps until the feature to be analyzed has a zero in the first row of *BM* (see Step 3c). Notice that the process of generating candidate feature lists and removing features from the lists is recursive (*TL* acts as a stack in which feature lists are added or removed in order to be reused in the analysis of new feature combinations).

The proposed method is described as follows:

```
Input: A basic matrix BM.
Output: The set of all typical testors of BM.
1. Sorting rows and columns of BM:
   a. Let F be the set of rows that have the minimum number of 1's.
   b. For each row $f \in F$ obtain the number of 1's in all columns of BM that contain a 1 in f. Put the row
      with the maximum number as the first row in BM. If there is more than one row with the maximum
      value, then take any one of them.
   c. Let $C^1$ ($C^0$) be the set of columns with a 1 (0) in the first row of BM. Rearrange the columns such
      that columns in $C^1$ are on the left and columns in $C^0$ are on the right. Sort, in descending order,
      the columns in $C^1$ according to its number of 1's. The columns in $C^0$ are sorted in the same way.
2. Initialization:
   a. L = []
   b. Let TTR be the list of typical testors, TTR = []. Notice that TTR is a list of lists.
   c. Let R be the list of all features in BM and TL = [R]. Notice that TL is also a list of lists.
3. Process:
   a. Let RL be the last list of features in TL, i.e. RL = last(TL).
   b. Let X be the first feature of RL.
   c. If | TL | = 1 then
        If the column corresponding to X ($c_X$) has a zero in the first row of BM, then return TTR and END
        else, if $c_X$ = (1,...,1) then TTR = TTR + [ [X] ], RL= RL \ [X] and go to Step 3b.
   d. L = subst(L, X, |TL|)
   e. Remove the last element (list) from TL, i.e. TL = TL \ [last(TL)].
   f. RL = RL \ [X]
   g. LP = nonExcl (RL, L)
   h. TTR = TTR + TypL(RL, L)
   i. If |RL| > 1, then
        TL= TL+ [RL]
        If | LP | > 1, then TL= TL + [LP]
   j. Go to Step 3.
```

Notice that the list *LP* includes the features of *RL* that are non-exclusive with *L* but do not form typical testors with *L,* whereas *TypL*(*RL*, *L*) contains the features of *RL* that constitute typical testors with the features in *L*. Then, the Steps 3g and 3h can be performed simultaneously as follows:

```
For each feature $X'$ of $RL$:
  1. Calculate $am_{L+[X']}$ from $am_L$ by using the Proposition 2.
  2. If $am_{L+[X']} \neq am_L$ (see condition 1 of Proposition 4) then
     a. Calculate $cm_{L+[X']}$ from $cm_L$ by using the Proposition 3.
     b. If $cm_{L+[X']} \wedge cx_i \neq (0,...,0)\ \forall X_i \in L$ (see condition 2 of Proposition 4) then
          If $am_{L+[X']} = (1,...,1)$ then
             Add $L+ [X']$ to $TTR$ ($L + [X']$ is a typical testor by Proposition 5)
          else, Add $X'$ to $LP$
```

The characteristics that allow *BR* method to avoid the analysis of irrelevant feature subsets are the following:

- The algorithm directly examines the feature combinations generated from the features in *L* and those belonging to *LP* (non-exclusive ones with *L*), avoiding the analysis of the remaining combinations.
- Since features that constitute a typical testor with *L* are not included in *LP*, the algorithm disregards all supersets of a typical testor.

CT-EXT method firstly generates testors and secondly verifies the typicity property, whereas *LEX* and *BR* methods firstly generate feature subsets satisfying the typicity property and then, verify the testor condition. *CT-EXT* attempted to reduce the cost of verifying the typicity property of *LEX*, but at expense of generating a greater number of feature subsets.

Example 2. The following table shows the feature subsets generated by *LEX*, *CT-EXT* and *BR* methods for the basic matrix *BM* until the subset $\{X_2,X_6\}$ (represented as 26 in the table) is generated. Typical testors are highlighted in boldface. As we can see, *BR* generates the least number of feature subsets. Notice that LP=[X_3,X_4] when the first 6 subsets are generated. Therefore, *BR* can jump to $\{X_1,X_3,X_4\}$ disregarding the remaining combinations that include X_1. However, *LEX* is not able to disregard these combinations. On the other hand, *CT-EXT* examines several subsets including X_1 and X_2 even though neither of them constitutes a typical testor. The definition of that a feature *X* contributes to a subset [8] only verifies that *X* has at least a typical row, but disregards that features in the subset can lose its typical rows due to *X*. Notice that X_2 contributes to $\{X_1\}$, but X_1 lost its typical row (the first one) due to X_2.

LEX		*CT-EXT*			*BR*	
1	146	1	135	**23**	1	**23**
12	**15**	12	136	**25**	12	24
13	**16**	123	14	**26**	13	**25**
134	2	124	145		14	**26**
135	**23**	125	146		**15**	
136	24	126	**15**		**16**	
14	**25**	13	**16**		**134**	
145	**26**	**134**	2		2	

$$BM = \left(\begin{array}{cccccc} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right)$$

4 Experimental Results

In order to evaluate the performance of the proposed method, we compare the time spent to obtain all typical testors by our method and the two best algorithms reported in the literature: *LEX* and *CT-EXT*. It is worth mentioning that the source code of the *CT-EXT* algorithm was provided by the authors. To ensure a fair comparison all methods are carried out on an Intel Pentium Dual Core 1.6 GHz, 1 Gb RAM.

For this comparison we use five data sets obtained from UCI Machine Learning Repository[1]. For each one, we generated the basic matrices regarding the strict equality as comparison criterion for all features. Table 1 shows the run time of the methods for the basic matrices of real data sets and two basic matrices randomly generated. Notice that these matrices have different dimensions (see Column 3). The last column (NTT) indicates the number of calculated typical testors.

Table 1. Run times (h:m:s:ms) of the algorithms for several basic matrices

Data set	Class	*BM*	*LEX*	*CT-EXT*	*BR*	NTT
Zoo (101 x 17)	7	14 x 17	0:0:00:15	0:0:0:718	0:0:00:00	34
Mushroom (8124 x 22)	2	30 x 22	0:0:00:16	0:0:0:750	0:0:00:00	292
Chess (3196 x 36)	2	29 x 36	0:2:22:16	0:8:01:67	0:0:00:12	4
Dermatology (366 x 34)	6	1124 x 34	0:25:45:7	1:43:15:6	0:0:58:22	115556
Promoter (106 x 57)	2	2761 x 57	1:07:27:5	4:24:23:8	0:3:18:51	7456943
Random	-[2]	150 x 70	0:55:45:3	2:06:30:4	0:4:02:67	44165054
Random	-[2]	100 x 100	2:22:01:9	> 20 hrs	0:10:30:1	183051234

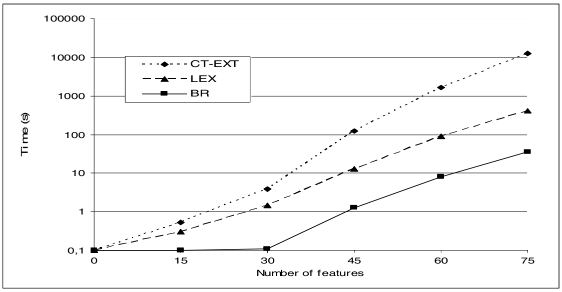


Fig. 1. Run times (in seconds) for basic matrices of 50 rows varying the number of features

As we can observe, the higher dimension of basic matrix, the greater time is needed to calculate typical testors in all methods. It is important to notice also that *BR* achieves considerable time reductions with respect to *LEX* and *CT-EXT*. Unlike the reported results in [8], our experiments revealed that *CT-EXT* actually performs worse than *LEX*.

In order to study the behavior of the algorithms, we show in Figure 1 the run times (in seconds) of the methods for basic matrices of 50 rows varying the number of

[1] http://archive.ics.uci.edu/ml/

[2] The number of classes is disregarded, because we randomly generate matrices of 0's and 1's.

features from 15 to 75. As we expected, the time of all methods grows exponentially when the number of features is increased. However, notice that our method runs about 10 times faster than the best competitor, *LEX* and about 100 times faster than *CT-EXT*.

Thus, we can conclude that *BR* is significantly more efficient than other algorithms.

5 Conclusions

In this paper, a new external scale algorithm *BR* to calculate all typical testors of a training matrix has been proposed. The experimental results demonstrate that this method significantly outperforms the two best algorithms reported in the literature. The main contributions that ensure the speed up in the calculation of the set of all typical testors are: a new method for verifying typicity and testor properties which is based on binary logic and profits from the computer bit operations, the introduction of a generation mechanism of candidate feature subsets that avoids the analysis of a greater number of irrelevant subsets, and a prior ordering of the basic matrix that guarantees that the method finishes as soon as possible.

Future work includes extending our method in order to obtain other generalizations of the typical testors not restricted to Zhuravlev's testors (e.g. ε-testors and fuzzy testors [2]). We also plan to conduct additional experiments with basic matrices of different densities to evaluate the performance of the proposed method.

References

1. Martínez-Trinidad, J.F., Guzmán-Arenas, A.: The Logical Combinatorial approach to Pattern Recognition: an overview through selected Works. Pattern Recognition 34(4), 741–751 (2001)
2. Lazo-Cortés, M., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An overview of the evolution of concept testor. Pattern Recognition 34(4), 753–762 (2001)
3. Ortiz-Posadas, M.R., Martínez-Trinidad, J.F., Shulcloper, J.R.: A new approach to diferential diagnosis of diseases. Int. J. Biomed. Compu. 40(3), 179–185 (1996)
4. De la Vega-Doria, L.A., Carrasco-Ochoa, J.A., Shulcloper, J.R.: Fuzzy KORA-W algorithm. In: 6th European Congress on Intelligent Techniques and Soft Computer, Aachen, Germany, pp. 1190–1194 (1998)
5. Pons-Porrata, A., Gil-García, R., Berlanga-Llavori, R.: Using Typical Testors for Feature Selection in Text Categorization. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 643–652. Springer, Heidelberg (2007)
6. Pons-Porrata, A., Ruiz-Shulcloper, J., Berlanga-Llavori, R.: A Method for the Automatic Summarization of Topic-Based Clusters of Documents. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 596–603. Springer, Heidelberg (2003)
7. Santiesteban-Alganza, Y., Pons-Porrata, A.: LEX: a New Algorithm for Computing Typical Testors. Revista Ciencias Matemáticas 21(1), 85–95 (2003)
8. Sánchez Díaz, G., Lazo Cortés, M.: CT-EXT: An Algorithm for Computing Typical Testor set. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 506–514. Springer, Heidelberg (2007)

Classifier Selection in a Family of Polyhedron Classifiers

Tetsuji Takahashi, Mineichi Kudo, and Atsuyoshi Nakamura

Division of Computer Science
Information Science and Technology
Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo 060-0814, Japan
{tetsuji,mine,atsu}@main.ist.hokudai.ac.jp
http://prml.main.ist.hokudai.ac.jp

Abstract. We consider an algorithm to approximate each class region by a small number of convex hulls and to apply them to classification. The convex hull of a finite set of points is computationally hard to be constructed in high dimensionality. Therefore, instead of the exact convex hull, we find an approximate convex hull (a polyhedron) in a time complexity that is linear in dimension. On the other hand, the set of such convex hulls is often too much complicated for classification. Thus we control the complexity by adjusting the number of faces of convex hulls. For reducing the computational time, we use an upper bound of the leave-one-out estimated error to evaluate the classifiers.

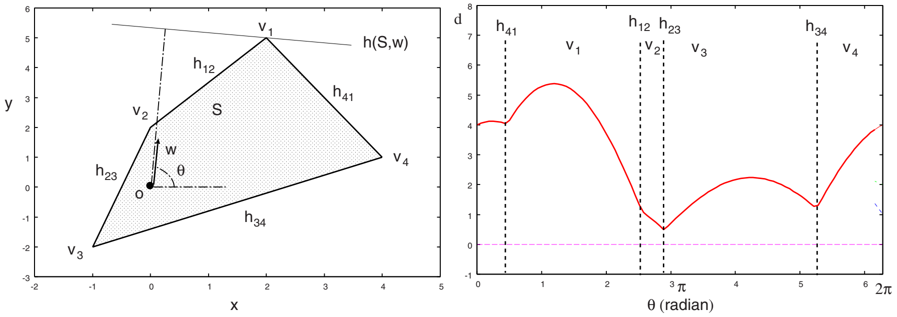
1 Introduction

In learning of classifiers, it is one promising way to estimate each class region by a set of convex hulls including samples of the class only. Then we can assign a class label to a given sample according to the distances of it to the estimated class regions. Indeed, such an approach [1] using convex hulls showed a comparable performance with SVM (Support Vector Machine) in low dimensionality [2]. In this paper, we discuss another way of approximating class regions by a set of polyhedral convex sets which are found in a linear order of dimension.

We can make the training error zero by using a sufficient number of convex hulls in such the way that every training sample of a class is covered by at least one convex hull and any other sample belonging to the other classes is excluded, as long as no sample is shared by more than one class. Thus, according to Occam's razor, we should choose the simplest classifier as long as it attains the same degree of training error. In our case, we may select the smallest number of convex hulls with the smallest number of faces.

There are three problems to be solved. First, in high dimensions, it costs too much to construct the convex hull of a given set of samples. Second, more than one convex hull is generally needed to cover all training samples exclusively. Finally, when we use the convex hulls for classification, it is required to measure the distance between a point and the boundary of each convex hull. In this paper,

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 441–448, 2009.



(a) (b)

Fig. 1. (a) Support plane $h(S, \boldsymbol{w})$ for a directional vector $\boldsymbol{w}$ with angle θ and the convex hull of S when all support planes are used. (b) The support function $d = H(S, \boldsymbol{w})$ with changing $\boldsymbol{w}(\theta)$.

we describe first a method, recently developed by us, that can cope with these three problems in a reasonable way [3]. Then a model selection procedure will be given as the contribution of this paper.

2 Definition of the Convex Hull and Reflective Convex Hull

In this paper, we use *support functions* for defining the convex hull of a set of points S [4]. A support function of a unit vector $\boldsymbol{w}$ ($\|\boldsymbol{w}\| = 1$) is given by

$$H(S, \boldsymbol{w}) = \sup\{\langle \boldsymbol{x}, \boldsymbol{w} \rangle | \boldsymbol{x} \in S\},$$

where "sup" denotes the supremum and "$\langle \cdot, \cdot \rangle$" denotes the inner product. With the set W_0 of all possible unit vectors, the convex hull C is defined as

$$C = conv(S, W_0) = \bigcap_{\boldsymbol{w} \in W_0} \{\boldsymbol{y} | \langle \boldsymbol{y}, \boldsymbol{w} \rangle \leq H(S, \boldsymbol{w})\}.$$

Here, $h(S, \boldsymbol{w}) = \{\boldsymbol{y} | \langle \boldsymbol{y}, \boldsymbol{w} \rangle = H(S, \boldsymbol{w})\}$ is called a *support plane*. The convex hull is an area which is surrounded by support planes $h(S, \boldsymbol{w})$. An example of the convex hull constructed by support planes is shown in Fig. 1.

We notice that a finite subset $W \subset W_0$ gives an approximate convex hull $conv(S, W)$ and thus a good selection of W derives a good approximation.

Next, let us consider separating a finite set S from another finite set T by the convex hull of S when they are linearly or non-linearly well-separated. A support plane of S might locate both S and T in the same side of the half-spaces specified by it. Apparently such support planes are useless for separating S from T. For separation of S from T, all we need is only *reflective support planes* which are support planes separating S from T perfectly or partly. A *reflective convex hull*,

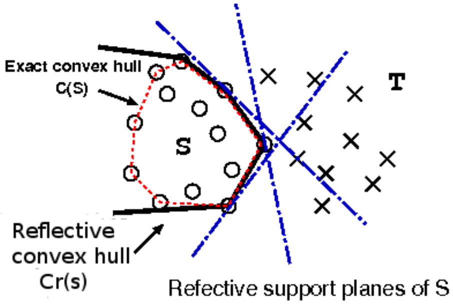


Fig. 2. The reflective convex hull of the set of positive samples S against the set of negative samples T. A few reflective support planes are shown.

$C_r = conv(S, W_r)$, is the polyhedral convex set specified by the set W_r of all unit vectors generating reflective support planes. Then the reflective convex hull C_r is formally defined by

$$C_r = conv(S, W_r) = \bigcap_{\boldsymbol{w} \in W_r} \{\boldsymbol{y} | \langle \boldsymbol{y}, \boldsymbol{w} \rangle \leq H(S, \boldsymbol{w})\}.$$

From the definition, $C = conv(S, W_0) \subseteq C_r = conv(S, W_r)$ since $W_r \subseteq W_0$. That is, the reflective convex hull of S is the polyhedral convex set of S whose faces reflect rays emitted from points in T. An example of the reflective convex hull is shown in Fig.2. Note that usually a reflective convex hull is not bounded unlike the convex hull.

We can also define the *margin* $M(S, T, \boldsymbol{w})$ between S and T in direction $\boldsymbol{w}$ as

$$M(S, T, \boldsymbol{w}) = -H(T, -\boldsymbol{w}) - H(S, \boldsymbol{w}).$$

Note that when S and T are linearly separable, then there exists a support plane specified by $\boldsymbol{w}$ with positive margin $M(S, T, \boldsymbol{w}) = M(T, S, -\boldsymbol{w})$. Now a reflective support plane can be defined as a support plane $h_r(S, \boldsymbol{w})$ satisfying

$$H(T, \boldsymbol{w}) - H(S, \boldsymbol{w}) > 0.$$

The (signed) distance between a point $\boldsymbol{x}$ and the nearest boundary of a convex hull $conv(S, W_0)$ is given by

$$D(\boldsymbol{x}, \partial conv(S, W_0)) = \sup_{\boldsymbol{w} \in W_0} \{M(S, \{\boldsymbol{x}\}, \boldsymbol{w})\}.$$

Here D takes a positive value for $\boldsymbol{x}$ outside of $conv(S, W_0)$ and a negative value for $\boldsymbol{x}$ strictly inside of $conv(S, W_0)$. The closer it is, the smaller the absolute value is. Note that, the calculation problem of $D(\boldsymbol{x}, \partial conv(S, W_0))$ is known to be NP-hard when $\boldsymbol{x}$ is inside of the convex hull, but if $W \subset W_r$ is finite, we can calculate the distance $D(\boldsymbol{x}, \partial conv(S, W))$ in a linear order of $|W|$ as

$$D(\boldsymbol{x}, \partial conv(S, W)) = \max_{\boldsymbol{w} \in W} \{M(S, \{\boldsymbol{x}\}, \boldsymbol{w})\}.$$

3 Approximation of a Class Region by the Reflective Convex Hulls

3.1 Algorithm

The algorithm [3] for each class is given as follows:

1. Let S be the positive sample set of a target class and T be the negative sample set of the other classes. Let $\mathcal{C} = W = \emptyset$. Let L be an upper bound of the number of convex hulls and K be the number of normal vectors.
2. Find random K pairs of $\boldsymbol{x}$ ($\in S$) and $\boldsymbol{y}$ ($\in T$) and put $\boldsymbol{w} = \frac{\boldsymbol{y}-\boldsymbol{x}}{\|\boldsymbol{y}-\boldsymbol{x}\|}$ in set W.
3. Repeat L times the following Steps 4–5.
4. Let $U = \emptyset$. According to a random presentation order of positive samples, add a positive sample $\boldsymbol{x}$ to U as long as $conv(U \cup \{\boldsymbol{x}\}, W) \cap T = \emptyset$.
5. Add the obtained $conv(U, W)$ into $\mathcal{C}$, unless it is already in $\mathcal{C}$.
6. Select a minimal subset of $\mathcal{C}$ by a greedy set cover procedure for all positive samples.

By this procedure, we have at most L approximated convex hulls that include samples of one class only (e.g. see Fig 3 (a)). It should be noted that each convex hull includes the positive samples maximally. We carry out this procedure twice in each class: the first one is for finding noisy samples that are included in only small convex hulls, and the second one is for obtaining the final convex hulls after removal of those noisy samples. To judge a noisy sample, we use a threshold θ. If a convex hull is small in size less than θ, that is, if the ratio of the number of samples included in the convex hull to the number of positive samples is less than $\theta(= 1\%$ in following experiments), all samples included in it are noisy. To emphasize the number of faces, an approximated reflective convex hull with K directional unit vectors is called a *K-directional approximated reflective convex hull* (shortly, K-ARCH). A convex hull might have less than K faces, but we use this terminology whenever $|W| = K$. Roughly speaking, as K increases, the corresponding K-ARCH approaches to the true reflective convex hull. The class assignment of an unknown sample is carried out on the basis of distance to the nearest boundary of K-ARCHs.

4 Classifier Selection

In reference [3], we used a fixed value of K for each dataset. In this paper, we choose a suboptimal value of K.

4.1 The Estimation of Generalization Error

As a measure of testing error, we use the LOO (Leave-One-Out) error rate. As well-known, LOO rate is almost unbiased, but it requires to build n classifiers for n samples. Hence, we consider an upper bound of LOO that can be obtained without reconstruction of classifiers. Let ϵ_{LOO} be the LOO error rate and V be

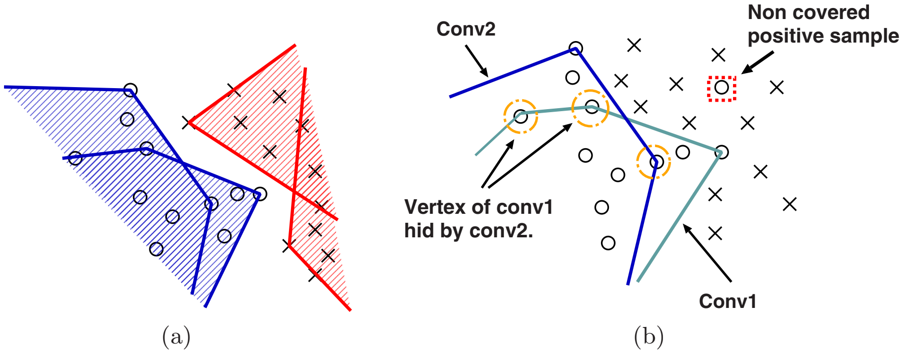


Fig. 3. (a) The approximated class region by reflective convex hulls. (b) Vertices necessary for LOO upper bound. The circled vertices are not necessary.

the set of vertices of all convex hulls and Z be the set of samples which are outside of all convex hulls. If a single convex hull is taken in each class, ϵ_{LOO} is bounded by the sum of $|V|$ and $|Z|$ decided by n. However, in the case that more than one convex hull is taken in a class, a vertex of one convex hull can be hidden by another. Fig.3 (b) illustrates such a case. Such vertices are able to be safely removed from the calculation. We call the other vertices "*effective vertices*" Let V' be the set of effective vertices on the boundary of the approximated class region. Then we have an upper bound by

$$\epsilon_{LOO} \leq \frac{|V'| + |Z|}{n}. \tag{1}$$

We use the value of the right-hand side of (1). Clearly there exists a trade-off between $|V'|$ and $|Z|$. So, we use the right-hand side term of (1) as our criterion.

4.2 Experiments

To construct W of normal vectors, We used n_p positive samples and $n_p(c-1)$ negative samples, so that $K = n_p^2(c-1)$ unit vectors were chosen randomly, where c is the number of classes. In following, we changed the value of n_p in $[1, 50]$, thus, K in $[c-1, 2500(c-1)]$.

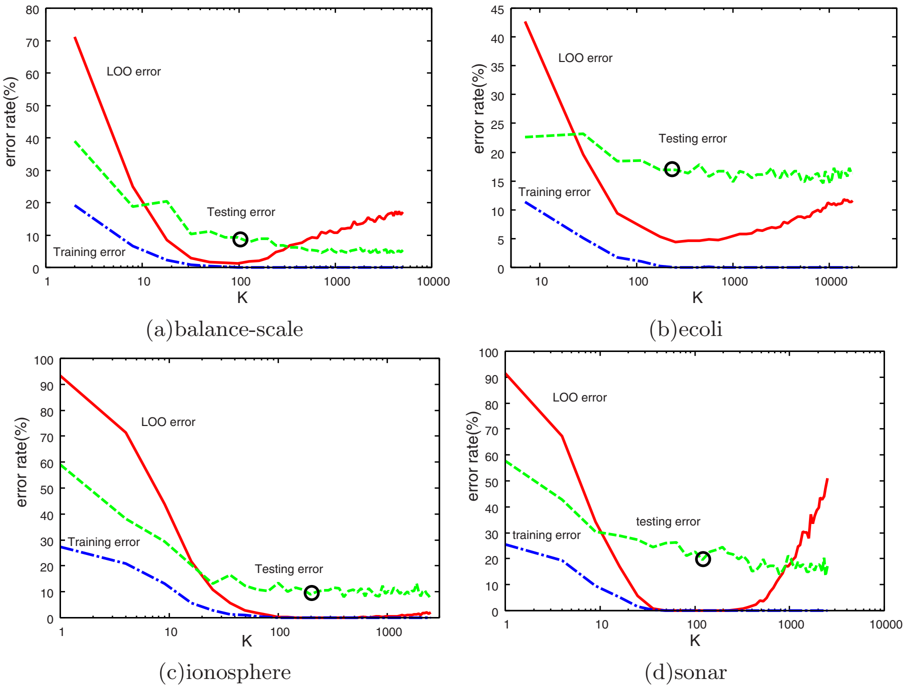
We used 9 datasets taken from UCI machine learning repository [5]. We increased the value of K until K reaches the maximum value. In each value of K, we repeated the algorithm 10 times for reducing the effect of the other random factors. Among 10 trials in a fixed value of K, we chose the best case in which the LOO estimate takes the minimum. The recognition rate was estimated by 10-fold cross validation. The loop number L of the randomized subclass method was set to $L = 20$. That is, the number of convex hulls was limited to 20 in each class.

4.3 Results

We compared the proposed K-ARCH algorithm with an SVM in which an RBF kernel with the default values of parameters (the standard deviation is $\sigma = 10.0$

Table 1. The recognition rates of SVM and K^*-ARCH where K^* is optimal in our model selection criterion and $|V'|$ is the number of effective vertices

Dataset	Classifier		#SV or $\lvert V' \rvert$	
	SVM	K^*-ARCH (K^*)	SVM	K^*-ARCH
balance-scale	**93.2**	90.3 (98)	255.0	200.1
diabetes	64.1	**75.0** (2401)	1310.0	483.3
ecoli	79.8	**83.0** (252)	385.7	144.6
glass	**66.3**	63.6 (180)	336.9	138.0
heart-statlog	59.3	**63.7** (2401)	479.4	201.8
ionosphere	**94.0**	90.9 (196)	132.5	331.2
iris	**98.0**	95.3 (18)	54.0	20.2
sonar	77.4	**80.4** (121)	214.0	429.3
wine	72.5	**87.0** (3200)	447.2	67.7
average	78.3	**81.0**	401.6	224.0



(a)balance-scale

(b)ecoli

(c)ionosphere

(d)sonar

Fig. 4. The error rate of K-ARCH as the number K of faces increases on four datasets. The three curves show the estimated LOO error (the right-hand term of Ineq. (1)), the training error and the testing error. The circled testing-error corresponds to the value of K^*.

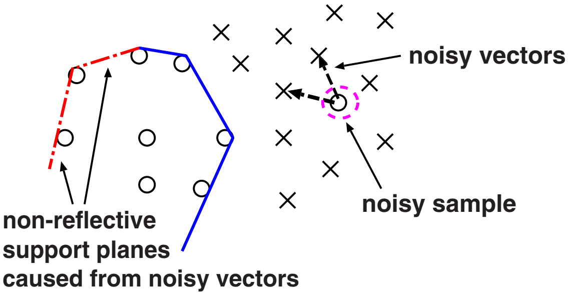


Fig. 5. Redundant faces generated by noisy vectors

and the soft margin parameter $\gamma = 100.0$) was used [2]. For K-ARCH algorithm, we use K^*-ARCHs for classification, where K^* is the value of K attaining the minimum LOO error. The result is shown in Table 1.

In Table 1, it is noted that K^*-ARCH performs better than half cases (5/9). It was worse for easier or well-separated class problems including `balance-scale`, `ionosphere` and `iris` (for these problems, the maximum recognition rate is over 90%). This might mean that K-ARCH tends to generate a little more complicated decision boundary compared with that of SVM. Note that a large number K^* is chosen for harder problems. It implies that K-ARCH formed a complex boundary. Note also that the number of (effective) vertices is often less than the number of support vectors. It means that K^*-ARCH often has higher sparsity than SVM.

We can see the details in some datasets in Fig. 4. From Fig. 4, we see that after reaching at the optimal value K^*, the testing error is not significantly reduced anymore. In general, a model selection criterion is expected to form a valley to simulate the testing error, but this is not the case. This implies that K-ARCH does not change its decision boundary even if the model becomes more complicated than necessary. We can interpret this phenomena as follows. Even if the faces increase more than necessary, but they are limited in the location opposite to the decision boundary. Such a situation is illustrated in Fig 5. As shown in Fig 5, such a redundant non-reflective support plane can be generated by some noisy samples. In this sense, we have to be careful about the value of θ used for the judgement of noisy samples. The curve of the testing error goes up and down to some extent as K increases. This is because small convex hulls with very acute angles can be generated when K is large.

5 Discussion

The algorithm K-ARCH needs a relatively high cost in data number n for constructing the classifiers. However, the complexity grows only linearly in the feature number m. The cost of finding polyhedral regions is $O(Kn^2m)$. To have one K-ARCH, we need $O(n_pn_n)$ for n_p positive and n_n negative samples. That high cost prevented us from dealing with larger datasets. However, for distance calculation of $D(\boldsymbol{x}, \partial conv_(U, W))$, we need only calculation of $O(Km)$.

The merit of our K-ARCH approach is that we can have several useful pieces of information in the original feature space unlike SVM. For example, the maximum margin in the corresponding reproducing kernel space does not always mean the maximum margin in the original space. So, the margin should be considered in the original space. It is well known for two linearly separable classes that the hyper-plane of SVM is equivalent to the bisector between the closest points on the boundaries of the convex hulls of those two classes [6]. Recently a similar relationship was revealed even for soft-margin SVM using the *reduced convex hulls* [7]. Our classifier is almost identical to SVM when two classes are linearly separable. In addition, our approaches use more than one convex hull in each class. It maximizes the margin locally even when two classes are non-linearly but smoothly separable. In this respect, our classifier is one of large-margin classifiers.

6 Conclusion

In this paper, a model selection method has been proposed for a family of polyhedron classifiers. The family is based on polyhedral class regions close to the convex hulls of some parts of training samples. In the family, the complexity mainly comes from the number of faces and vertices of each polyhedral region. The selection method employed an upper bound of the LOO error for time reduction and showed a satisfactory result for model selection.

References

1. Kudo, M., Shimbo, M.: Appriximation of class region by convex hulls. Technical report of IEICE. PRMU 100, 1–6 (2000) (in Japanese)
2. Collobert, R., Bengio, S.: SVMTorch: support vector machines for large-scale regression problems. Journal of Machine Learning Research 1, 143–160 (2001)
3. Kudo, M., Takigawa, I., Nakamura, A.: Classification by reflective convex hulls. In: Proceedings of 19th International Conference on Pattern Recognision (ICPR 2008), Tampa, Florida, USA (2008)
4. Ghosh, P., Kumar, K.: Support function representation of convex bodies, its application in geometric computing, and some related representations. Computer Vision and Image Understanding 72, 379–403 (1998)
5. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
6. Zhou, D., Xiao, B., Zhou, H.: Global geometry of svm classifiers. Technical report, AI Lab, Institute of Automation, Chinese Academy of Sciences (2002)
7. Theodoridis, S., Mavroforakis, M.: Reduced convex hulls: A geometric approach to support vector machines. IEEE Signal Processing Magazine 1 (2007)

Characterisation of Feature Points in Eye Fundus Images

D. Calvo, M. Ortega, M.G. Penedo, and J. Rouco

VARPA Group, Department of Computer Science, University of A Coruña, Spain
{dcalvo,mortega,mgpenedo,jrouco}@udc.es

Abstract. The retinal vessel tree adds decisive knowledge in the diagnosis of numerous opthalmologic pathologies such as hypertension or diabetes. One of the problems in the analysis of the retinal vessel tree is the lack of information in terms of vessels depth as the image acquisition usually leads to a 2D image. This situation provokes a scenario where two different vessels coinciding in a point could be interpreted as a vessel forking into a bifurcation. That is why, for traking and labelling the retinal vascular tree, bifurcations and crossovers of vessels are considered feature points. In this work a novel method for these retinal vessel tree feature points detection and classification is introduced. The method applies image techniques such as filters or thinning to obtain the adequate structure to detect the points and sets a classification of these points studying its environment. The methodology is tested using a standard database and the results show high classification capabilities.

Keywords: Feature points, classification of features, retinal images.

1 Introduction

In the field of medical diagnosis and disease study, it is necessary to analyse in detail medical images. This analysis usually covers the measuring of parameters, the calculation of values according to the image, and the monitoring of the structures. These tasks are usually performed manually by experts. This specialised process, takes up a lot of time and, as the task is done manually, is sensitive to subjective errors. It is, therefore, necessary to use more reliable methods.

The vascular tree of the retina can show morphological variations due to diseases or even aging. The branches intertwine, creating points where several vessels coincide. These points are of special importance in terms of analysis of the tree as, depending whether they are in the same spatial plane or not, they can be physically connected or otherwise just appear to be, due to the perspective of the image. These will be the feature points in this work.

Many methods for extracting information from the retina vessel tree can be found in the literature, but authors usually limit their work to a two dimensional extraction of the information. An analysis of the third dimension, depth, is needed. In the bibliography there are some works that try to solve this problem. For instance, the work proposed by Ali Can *et al.* [1] tries to solve the problem

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 449–456, 2009.

in difficult images using the central vessel line to detect and classify the feature points. Other methods, like the proposed by Chia-Ling Tsai *et al.* [2], use vessel segments intersections as seeds to track vessel centre lines and classify feature points according to intersection angles. The work proposed by Enrico Grisan *et al.* [3] extracts the structure using a vessel tracking based method needing a previous step before detecting feature points to fix the loss of connectivity in the intersections. Another work, the proposed by V.Bevilacqua *et al.* [4], uses a small window to analyse the whole skeleton of the vascular structure. The main problem of this solution is the misclassification of crossovers, as they are only properly classified when the vessel segments intersect exactly in the same pixel.

This paper proposes a method to detect feature points of the retinal vascular tree and a subsequent classification of the detected points in two classes, bifurcations and crossovers. From an image of the retinal structure of the eye, the vascular tree is segmented. From this segmentation the skeleton is obtained, where the feature points are detected. In the last step, these feature points are classified according to a local analysis and a topological study.

The paper is organised as follows: in section 2 the segmentation process is presented. Section 3 describes the feature point detection method. In section 4 a description of the classification method used is presented. Section 5 shows the experimental results and validation obtained using standard retinal image databases. Finally, section 6 provides some discussion and conclusions.

2 Arteriovenous Structure Segmentation

Feature point detection implies an analysis of the vascular structure so a segmentation of the retinal vessel tree is required. In this work we use an approach with a particularly high sensitivity and specificity at classifying points as vessel or non vessel points. This segmentation process is done in two main steps: vascular structure enhancement and extraction of the arteriovenous tree.

By performing an initial enhancement the causes of a potential malfunction of the whole process, such as noise or vessel reflections are eliminated.

The preprocessing step applies a Tophat filter [5] to enhance the biggest and darkest structures present in the image, corresponding the vessels. Then, a median filter is applied to reduce noise and to tone down vascular reflex.

The vessel enhancement step uses a multiscalar approximation where the eigenvalues of the Hessian matrix [6] are used to apply a filter process that detects different sized geometric tubular structures. A function $B(p)$ is defined to measure the membership of a pixel, p, to vessel structures:

$$B(p) = \begin{cases} 0 & \text{if} \quad \lambda_2 < 0 \\ exp(-2R_b^2)(1 - exp(-\frac{S^2}{2c^2})) & \end{cases} \tag{1}$$

where $R_b = \dfrac{\lambda_1}{\lambda_2}$ (one and two eigenvalue), c is the half of the max hessian norm, S represents a measure to "second order structures". Vessel pixels are characterised by a small λ_1 value and a higher positive λ_2 value.

Once the blood vessels are enhanced, the vascular extraction is done in two steps: first an early segmentation and, second, a removal of isolated pixels.

An hysteresis based thresholding is done in the segmentation task. A hard threshold (T_h) obtains the pixels with a high confidence of being vessel pixels while the weak threshold (T_w) keeps all the pixels of the tree, even spurious ones. The final segmentation will be formed by all the pixels selected by the weak threshold connected to, at least, one pixel obtained by the hard threshold. T_h and T_w are obtained from two image properties: the percentage of image representing vessels and the percentage of the image classified as fundus. The gap between both percentages will include all not classified pixels. After calculating the percentiles with Equation 2 obtaining the values for the thresholds is immediate.

$$P_k = L_k + \frac{k\left(\frac{n}{100}\right) - F_k}{f_k} * c, k = 1, 2, ..., 99 \tag{2}$$

where L_k is percentile k lower limit, n stands for the size of data set, F_k is the accumulated frequency for $k-1$, f_k represents the frequency of percentile k and c is the size of the percentile interval (1 in this case).

To be able to obtain adequate results not only in high quality images from healthy eyes but also in poor images or images from eyes with diseases a last step is taken to erase spurious structures that, not belonging to the vascular structure, reached this point. To solve this, all isolated structures smaller than a prefixed number of pixels are erased. Figures 1(a) and 1(b) show, respectively, the original and segmented image.

Segmentation method was validated over 40 images from DRIVE [7] database using a 15x15 window for the Tophat filter and reaching a precision of 95%.

3 Feature Point Detection

A feature point can be defined as every non internal vessel point. Specifically, feature points, are vessel crossovers, bifurcations or end points. The goal in this first stage is to detect the feature points of the retinal vessel tree. It is clear that, in the segmented image (Figure 1(b)), some properties are not constant along all the structure, like vessel width that decreases as the branch level of the structure becomes deeper. To unify this property, it is needed a method to reduce vessel width to one pixel without changing direction or connectivity. The skeleton is the structure that meets all these properties.

The results of the segmentation process force a previous preprocessing step before the skeletonization.Fig.1(b) shows gaps inside the segmented vessels that would give a wrong skeleton structure creating false feature points. To avoid these spurious feature points it is necessary to “fill” the gaps inside the vessels. For this, a dilation process using a modified median filter, to avoid erosion, is applied making the lateral vessel borders grow towards the centre filling the mentioned gaps. To “fill” as much white gaps as possible the dilation process is applied several times. The value of N depends on the spatial resolution of the

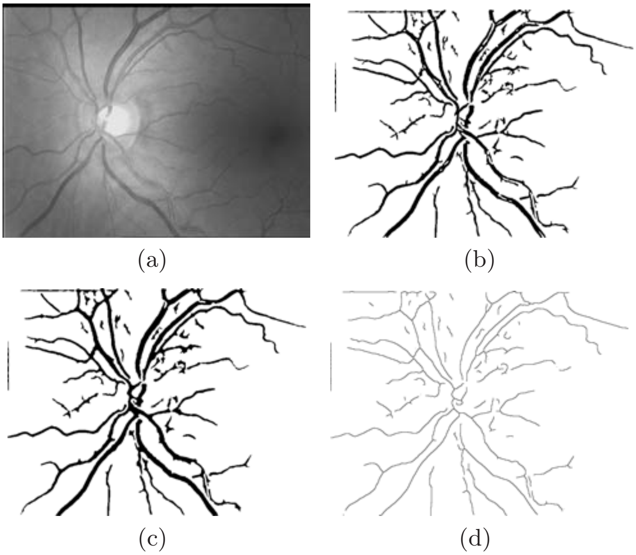


Fig. 1. (a) Original image,(b) segmented image , (c) result of the dilation process with $N = 4$ and (d) skeletonized vascular tree

images used, with the images used in this work (768x584) it was determined empirically that optimal values for N were around 4 (Fig.1(c)).

Representing the skeleton by the medial axis function(MAF), defined as the set of points centre of the maximum radius circles that fit inside the vessels, is a very heavy task so template based method, versatiles and effectives, are applied to the segmented image. In this work the Stentiford thinning method [8] is used. Fig.1(d) shows the skeletonization results.

3.1 Feature Points Location

The feature point location is done according to the local information of the points. so an analysis of the neighbours of each point is done. According to the intersection number, $I(V)$, calculated for each point,V, as shown in 3 each point will be marked as an end point when $I(V) = 1$, internal point when $I(V) = 2$ and crossover or bifurcation when $I(V) > 2$.

$$I(V) = \frac{1}{2}\left(\sum_{i=1}^{8} |N_i(V) - N_{i+1}(V)|\right) \tag{3}$$

where, $N_i(V)$ are the neighbours of the analysed point V named clockwise consecutively.

Skeletonization process forces a pruning step to erase small artificial branches that create spurious feature points. Branches, understood as vessel segments between an end point and another feature point, are tracked and erased if smaller than the maximum vessel width expected in the image, ζ.

For the evaluation of the feature point detection process a set of 30 images from VARIA [9] database is used. The system obtained a value of sensitivity of 89.7% while only detecting a total of 32 false positives for a total of 662 true positives.

4 Feature Point Classification

The method used in [4] for the classification between bifurcations and crossovers, similar to the one used in this work for the detection, can lead to an incorrect classification of crossovers when, due to the angle and width of interwining vessels, not all vessel segments coincide in the same skeleton pixel. To solve this problem and produce a robust and valid classification a further analysis, according to local and topological features, for points with $I(V) > 2$ is done.

The first classification step is done according to local features of the points without considering, for it, the effect of the other points. Each detected feature point, F, is used as centre of a circumference with radius R_c used for the analysis. $n(F)$ gives the number of vessel segments that intersect the circumference where $n(F) = 3$ corresponds to a bifurcation and $n(F) = 4$ to a crossover. Fig.2 shows the blood vessels, the circumference used to do the analysis, and, coloured darker, the pixels where the vessels intersect the circumference.

To avoid missclasifications when the circumference is intersected by vessels alien to the analysed point, a vote system with three radius is used. Two values, C(F) and B(F), meaning the number of votes for a point F to be classified, respectively, as a crossover and a bifurcation are used:

$$C(F) = 2 * C(F, R_1) + C(F, R_c) + C(F, R_2) \tag{4}$$

$$B(F) = B(F, R_1) + B(F, R_c) + 2 * B(F, R_2) \tag{5}$$

where $C(F, R_i)$ and $B(F, R_i)$ are binary values indicating if F is classified, respectively, as a crossover or a bifurcation using a radius R_i, $R_1 = R_c - \rho$ and $R_2 = R_c + \rho$, with ρ a fixed amount, are the radius around R_c. Note that the contribution of the small radius is more valuable, and therefore weighted, in the crossover classification while for bifurcations the big radius adds more information. F will be classified as a crossover when $C(F) > B(F)$ and a bifurcation otherwise.

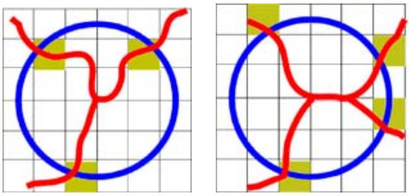

(a) (b)

Fig. 2. Preliminary feature point classification according to the number of vessel intersections where (a) represents a bifurcation and (b) a crossover

Due to the representation of crossovers in the skeleton, this information is not enough to assure that a feature point is a crossover while being a necessary condition. According to this, a topological classification is needed analysing the feature points in pairs, (F_1, F_2), attending to their Euclidean distance $d(F_1, F_2)$. If both F_1 and F_2 are connected by a vessel segment and $d(F_1, F_2) <= 2 * R_c$, both points are merged into a crossover in the middle point between them.

Not classified real crossovers would create two false bifurcations in the final result. Thus, another threshold, R_b, is needed to decide which points are accepted as bifurcations. For each pair of bifurcations, understood as two connected bifurcations that minimise their Euclidean distance, a circumference with radius R_b centred in the middle point between them is used. This circumference cannot contain both points. So, points not fulfilling the conditions are marked as not classified. Note that R_c and R_b parameters allow to tune the system in terms of specificity and sensitivity as some domains would require different performances. In the next section, some experiments and performance results are shown.

5 Results

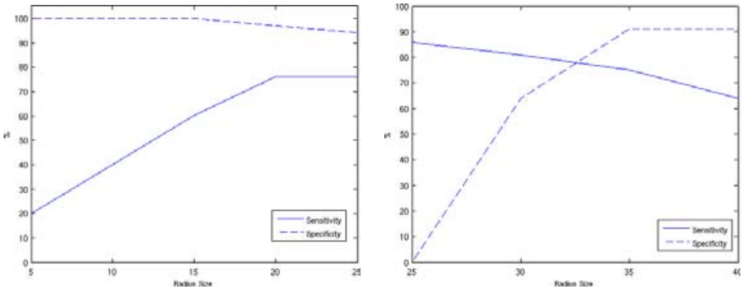
For the analysis of the methodology a set of 45 images randomly extracted from VARIA [9] and labelled by experts were used. These images are centred in the optic disc with spatial resolution of 768x584 pixels.

Image preprocessing parameters are necessary to the correct performance of later steps. For the image set used, the adequate number of dilations, N, is four, the chosen prune threshold is $\zeta = 20$ and the radius around R_c in the vote system, $\rho = 5$.

R_c and R_b allow to tune the specificity and sensitivity of crossover and bifurcation classification respectively, so a quantitative study according to these parameters is presented. The results allow to choose the adequate parameters for a specific domain where the desired sensitivity or specificity levels can change depending on the False Positives, True Positives and False Negatives as shown in Fig.3(a).

The figure shows how the number of correct classified crossovers increase with the radius size. This tendency could throw the idea of increasing the radius size until obtaining a big number of classified crossovers, however, increasing the radius also increases the number of misclassified crossovers.

Fig.3(b) shows the results for bifurcation classification according to the chosen R_b radius. This figure displays a new category, the non classified points, that includes the points that fulfilling the morphology conditions are not close enough to be classified as crossover and not far enough not to be classified as independent bifurcations. The bigger radius, R_b, used the more number of points without classifying but the number of false positives will be below the 1%. Opposite to this, if a big level of true positives is needed with a small radius the sensitivity is over the 70%. Selecting $R_c = 10$ and $R_b = 30$, the global sensitivity of the system is 75% and the specificity 93%.



(a) (b)

Fig. 3. Analysis of the influence of R_c for the crossovers, (a), and R_b for the bifurcations, (b) in the classification performance of the system

Table 1. Obtained results compared to the results given in [3]

	Bifurcations		Crossovers	
	Sensitivity	Specificity	Sensitivity	Specificity
D. Calvo *et al.*	75%	91%	76%	96%
E. Grisan *et al.*	76%	87%	62%	74%

As said in Section 3, the technique shown in [4] has the problem of the crossover misclassification due to the skeleton representation where a crossover turns into two close bifurcations. This paper, [4], does not offer quantitative results but our implementation of this technique shows that nearly every point is classified as a bifurcation, being capable to classify correctly only 3% of the crossovers. The work proposed in [3] extracts the structure using a vessel tracking based with the results shown in Table 1. Other previous techniques do not offer quantitative results in the characterisation task to compare with. The main improvement comes in the crossover rate, due to the radius proposed. In general, the system exhibits a very high specificity rate for both classes making it suitable for critical tasks.

6 Conclusions and Future Work

In this work a method for the detection and classification of the feature points of the retinal vascular tree using several image processing techniques has been presented. The detection and classification of these points is important because it increases the information about the retinal vascular structure. Having the feature points of the tree allows an objective analysis of the diseases that cause modifications in the vascular morphology.

To improve the system a future work could be use vessel features to classify the feature points. The classification method is done now according to the number of vessels that belong to the point and in the relationship between pairs of points. The presented work is able to be applied in many other domains such as authentication task, using retinal images in order to help the comparison between points according to the classification given by this method.

Acknowledgements

This paper has been partly funded by the Ministerio de Ciencia y Tecnología. Instituto de Salud Carlos III through the grant contract PI08/90420.

References

1. Can, A., Shen, H., Turner, J., Tanenbaum, H., Roysam, B.: Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. IEEE Transactions on Information Technology in Biomedicine 3(2), 125–138 (1999)
2. Tsai, C.L., Stewart, C., Tanenbaum, H., Roysam, B.: Model-based method for improving the accuracy and repeatability of estimating vascular bifurcations and crossovers from retinal fundus images. IEEE Transactions on Information Technology in Biomedicine 8(2), 122–130 (2004)
3. Grisan, E., Pesce, A., Giani, A., Foracchia, M., Ruggeri, A.: A new tracking system for the robust extraction of retinal vessel structure. In: IEMBS 2004. 26th Annual International Conference of the IEEE, September 2004. Engineering in Medicine and Biology Society, vol. 1, 3, pp. 1620–1623 (2004)
4. Bevilacqua, V., Cambó, S., Cariello, L., Mastronardi, G.: A combined method to detect retinal fundus features. In: Proceedings of IEEE European Conference on Emergent Aspects in Clinical Data Analysis (2005)
5. Dougherty, E.R.: Mathematical morphology in image processing / edited by Edward Dougherty. M. Dekker, New York (1993)
6. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering, 130+ (1998)
7. Drive: Digital retinal images for vessel extraction, http://www.isi.uu.nl/Research/Databases/DRIVE/
8. Stentiford, F.W.M., Mortimer, R.G.: Some new heuristics for thinning binary handprinted characters for ocr. IEEE Transactions on Systems, Man and Cybernetics 13(1), 81–84 (1983)
9. Varia: Varpa retinal images for authentication, http://www.varpa.es/varia.html

Combining Functional Data Projections for Time Series Classification

Alberto Muñoz and Javier González

Universidad Carlos III de Madrid, c/ Madrid 126, 28903 Getafe, Spain
{javier.gonzalez,alberto.munoz}@uc3m.es

Abstract. We afford the classification of time series in the Functional Data Analysis (FDA) context. To this aim we introduce projections methods for the time series onto appropriate Reproducing Kernel Hilbert Spaces (RKHSs) with the aid of Regularization Theory. Next we project the curves onto a set of different RKHSs. Then we consider the induced Euclidean metrics in these spaces and combine them in order to obtain a single kernel valid for classification purposes. The methodology is tested on some real and simulated classification examples.

Keywords: Functional data, Regularization Theory, Reproducing Kernel Hilbert Spaces, Kernel Combination, Classifier Fusion.

1 Introduction

The field of Functional Data Analysis (FDA) [12,6] deals naturally with data of very high (or intrinsically infinite) dimensionality. A typical example are time series early studied by Parzen [11]. In practice, a functional datum is given as a set of discrete measured values. FDA methods first convert these values to a function and then apply some generalized multivariate procedure able to cope with functions.

The standard way to reduce functional data dimension is to project the functional data onto some space of functions. This approach has been extensively studied, and many papers in FDA deal with the election of the best basis [12] of the space: Fourier analysis, Wavelets, B-splines basis and Functional Principal Component Analysis (FPCA) constitute some common examples.

The key idea in our proposal is to consider each function as a point in a given function space and then to project these points onto a set of some finite-dimensional function subspaces. Then, we define appropriate kernels for those projections and combine them to obtain a kernel function valid for classification purposes. To this aim, we consider several Mercer kernels and project the original time series onto the Reproducing Kernel Hilbert Spaces (RKHS) [1,15,9,3] associated to these kernels, obtaining different finite dimensional representations of the original series.

To achieve the goal of information fusion in this context, we need to obtain a single representation for the curves from the set of different representations. To this aim we consider the natural (Euclidean) kernel matrices that arise from the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 457–464, 2009.

obtained representations and fuse them using some kernel combination technique [5]. Then, we use the Kernel Fusion [10] to obtain the function kernel to be used to classify the time series using Support Vector Machines.

This work is organized as follows. In Section 2 we show how to project a set of curves onto a RKHS generated by the eigenfunctions of a given kernel with the aid of Regularization Theory. In Section 3 we fuse the information provided by the previous projection in the frame of kernel combinations theory. We illustrate the performance of the proposed combination theory for functional data in some simulated and real experiments in Section 5 and we outline some future research lines of research in Section 6.

2 Representing Functional Data in a Reproducing Kernel Hilbert Space

Let $\{\hat{c}_1, \ldots, \hat{c}_m\}$ denote the available sample of curves. Each sampled curve $\hat{c}_l$ is identified with a data set $\{(\mathbf{x}_i, \mathbf{y}_{il}) \in X \times Y\}_{i=1}^n$. X is the space of input variables and, in most cases, $Y = \mathbb{R}$. We assume that, for each $\hat{c}_l$, there exists a continuous function $c_l : X \longrightarrow Y$ such that $E[y_l|\mathbf{x}] = c_l(\mathbf{x})$ (with respect to some probability measure). Thus $\hat{c}_l$ is the sample version of c_l. Notice that, for simplicity in notation, we assume that the $\mathbf{x}_i$ are common for all the curves, as it is the habitual case in the literature [12].

There are several ways to introduce RKHS (see [9,1,4,15]). In a nutshell, the essential ingredient for a Hilbert function space H to be a RKHS is the existence of a symmetric positive definite function $K : X \times X \rightarrow \mathbb{R}$ named Mercer Kernel (or reproducing kernel) for H [1]. The elements of H, called H_K in the sequel, can be expressed as finite linear combinations of the form $h = \sum_s \lambda_s K(x_s, \cdot)$ where $\lambda_s \in \mathbb{R}$ and $x_s \in X$.

Consider the linear integral operator T_K associated to the kernel K defined by $T_K(f) = \int_X K(\cdot, s)f(s)ds$. If we impose that $\int\int K^2(x,y)dxdy < \infty$, then T_K has a countable sequence of eigenvalues $\{\lambda_j\}$ and (orthogonal) eigenfunctions $\{\phi_j\}$ and K can be expressed as $K(x,y) = \sum_j \lambda_j \phi_j(x)\phi_j(y)$ (where the convergence is absolute and uniform).

Given two function f and g in a function general space (that contains H_K as a subspace), they will be projected onto H_K using the operator T_K. Thus, the projections f^* and g^* will belong to the range of T_K, being $f^* = \Pi_{H_K}(f) = T_K(f)$ and $g^* = \Pi_{H_K}(g) = T_K(g)$. Applying the Spectral Theorem to T_K we get:

$$f^* = T_K(f) = \sum_j \lambda_j \langle f, \phi_j \rangle \phi_j, \quad g^* = T_K(g) = \sum_j \lambda_j \langle g, \phi_j \rangle \phi_j \tag{1}$$

Definition 1. *Let K a kernel function with eigenfunction $\{\phi_j\}$ and T_K the linear integral operator associated to K. Consider f and g two curves in a general space Ω containing H_K. Then, we define the* ***Spectral Inner Product*** *of f and g in Ω by:*

$$\langle f, g \rangle_\Omega = \langle \Pi_{H_K}(f), \Pi_{H_K}(g) \rangle_{H_K}, \tag{2}$$

Notice that $\langle f, g\rangle_\Omega = \langle f^*, g^*\rangle_{H_K} = \sum_j \mu_j \gamma_j$ *is the standard inner product of the two elements* $f^* = \sum_j \mu_j \phi_j$ *and* $g^* = \sum_j \gamma_j \phi_j$ *in* H_K*.*

Next we want to obtain c_l^* for each c_l (the function corresponding to the sample functional data point $\hat{c}_l \equiv \{(\mathbf{x}_i, y_{il}) \in X \times Y\}_{i=1}^n$) in order to have a practical way to estimate the projections of the curves and to calculate (1) and (2). To find the coefficients of c_l^* (in terms of the ϕ_j in eq. (1), we use Regularization Theory to express the approximation of $\hat{c}_l$ in terms of a kernel expansion. To this aim, we seek the function c_l^* that solves the following optimization problem [4], [9]:

$$\arg\min_{c \in H_K} \frac{1}{n} \sum_{i=1}^{n} L(y_i, c(\mathbf{x}_i)) + \gamma \|c\|_K^2 \,. \tag{3}$$

where $\gamma > 0$, $\|c\|_K$ is the norm of the function c in H_K, $y_i = \hat{c}_i$ and $L(y_i, c(\mathbf{x}_i)) = (c(\mathbf{x}_i) - y_i)^2$. Expression (3) measures the trade-off between the fitness of the function to the data and the complexity of the solution (measured by $\|c\|_K^2$). By the Representer Theorem [14], the solution c_l^* to the problem (3) exists, is unique and admits a representation of the form

$$c_l^*(\mathbf{x}) = \sum_{i=1}^{n} \alpha_{li} K(\mathbf{x}_i, \mathbf{x}), \quad \forall \mathbf{x} \in X \text{ where } \alpha_i \in \mathbb{R} \,. \tag{4}$$

where $\alpha_l = (\alpha_{il}, \ldots, \alpha_{nl})$ is the solution to the linear system $(\gamma n I_n + K_S)\alpha_l = y_l$ where $K_S = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j}$.

2.1 Functional Data Projections onto the Eigenfunctions Space

The particular projection we use in this work is given as follows:

Proposition 1. *Let* c *be a curve, whose sample version is* $\hat{c} = \{(\boldsymbol{x}_i, y_i) \in X \times Y\}_{i=1}^n$ *and* K *a kernel with eigenfunctions* $\{\phi_1 \ldots, \phi_d\}$ *(basis of* H_K*). Then, the projected curve* $c^*(\boldsymbol{x})$*, given by the minimization of (3), can be expressed as*

$$c_l^*(\boldsymbol{x}) = \sum_{j=1}^{d} \lambda_{lj}^* \phi_j(\boldsymbol{x}). \tag{5}$$

where λ_{lj}^* *are the weights of the projection of* $c^*(\boldsymbol{x})$ *onto the function space generated by the eigenfunctions of* K *(Span*$\{\phi_1 \ldots, \phi_d\}$*). In practice (where a finite sample is available)* λ_{lj}^* *can be estimated by*

$$\hat{\lambda}_{lj}^* = \hat{\lambda}_j \sum_{i=1}^{n} \alpha_{li} \hat{\phi}_{ji}, \tag{6}$$

being $\hat{\lambda}_j$ *the jth eigenvalue corresponding to the eigenvector* $\hat{\phi}_j$ *of the matrix* $K_S = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j}$, $d = \min(n, r(K_S))$*, and* α_{li} *the solution to problem (3). See [7] for further details.*

Thus we represent each curve c_l by the vector $\lambda_l = (\lambda_{l1}^*, ..., \lambda_{ld}^*)$. This representation has the nice property that is continuous in the input data [7].

3 Combining Projections via Kernel Combinations

In this section we show how to combine different representations of the curves given by the projections of the time series onto different spaces. To this aim we use some kernel combination technique to fuse the information of kernel matrices that arise from the obtained representations.

3.1 Kernels and Induced Projections

Let K a kernel function and c_1 and c_2 two time series with sample versions $\hat{c}_1$ and $\hat{c}_2$. Consider the particular curve projection onto H_K given by (3). In this case, the Spectral Inner Product of c_1 and c_2 is given by $\langle c_1, c_2 \rangle = \sum_j \lambda_{1j}^* \lambda_{2j}^*$ where λ_1^* and λ_2^* are the finite dimensional representation c_1 and c_2 in (5). Given that λ_j^* is never available we use its estimation given by eq. (6): $\sum_j \hat{\lambda}_{1j}^* \hat{\lambda}_{2j}^*$.

Definition 2. *Let $\{\hat{c}_1, \dots, \hat{c}_m\}$ a set of sample curves and K a kernel function. the* ***Spectral Kernel*** *(SK) induced by a kernel K for two sample curves $\hat{c}_l$ and $\hat{c}_t$ is defined by*

$$\tilde{K}(\hat{c}_l, \hat{c}_t) = (\hat{\lambda}_l^*)^T \hat{\lambda}_t^*, \tag{7}$$

for $\hat{\lambda}_l^* = (\lambda_{l1}^*, \dots, \lambda_{ld}^*)$ and $\hat{\lambda}_t^* = (\lambda_{t1}^*, \dots, \lambda_{td}^*)$ the representation of the curves l and t estimated by eq. (6).

3.2 Combining the Representations

Let $K_1, K_2, ..., K_p$ be a set of p kernels functions inducing p different RKHS $H_{K_1}, ..., H_{K_p}$ and let $S = \{\hat{c}_1, \dots, \hat{c}_m\}$ a labeled set of sample curves where each $\hat{c}_t$ is identified with a data set $\hat{c}_t = \{(\mathbf{x}_i, \mathbf{y}_{il}, z_l)\}_{i=1}^n$ with $z_t \in \{-1, 1\}$ (the labels of the curves). Let $\tilde{K}_{S1}, ..., \tilde{K}_{Sp}$ the p Spectral Kernels matrices (see eq. (7)) associated to the projections of the sample curves onto the spaces $H_{K_1}, ..., H_{K_p}$.

We want to combine the Spectral kernel matrices $\tilde{K}_{S1}, ..., \tilde{K}_{Sp}$ to obtain a single kernel function K^* that induces a single representation of the curves appropriate in classification problems. To this aim we select some functional kernel combination methods proposed in [9,5]. In particular we will use the Average Kernel Method (AKM), the Modified Average Kernel Method (MAKM), the Absolute Value Method (AV) and the Max-Min method. However the resulting combination matrix K^* does not need to be positive definite and does not allow directly to evaluate K_S^* at new points (where labels are not available). To fix simultaneously both problems we use the Fusion Kernel proposed in [10].

Definition 3 (Fusion Kernel). *Let $\tilde{K}_1, \dots, \tilde{K}_p$ be a set of p kernel functions (Spectral Kernels in our case). A kernel function K is a Fusion Kernel for the set $\tilde{K}_1, \dots, \tilde{K}_p$ when it can be expressed as*

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{h=1}^{d} \lambda_h \phi_h(\boldsymbol{x}) \phi_h(\boldsymbol{y}), \tag{8}$$

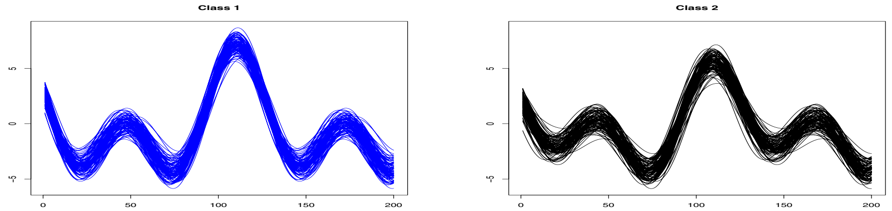


Fig. 1. Two classes of the simulated curves of the experiment

where $\{\lambda_h\} \in \mathrm{I\!R}^+$ *and* $\phi_h \in Span\langle \underbrace{\phi_{11}, \ldots, \phi_{1d_1}}_{\tilde{K}_1}, ..., \underbrace{\phi_{p1}, \ldots, \phi_{pd_p}}_{\tilde{K}_p} \rangle$ *and* ϕ_{jr} *represents the jth eigenfunction of the rth kernel.*

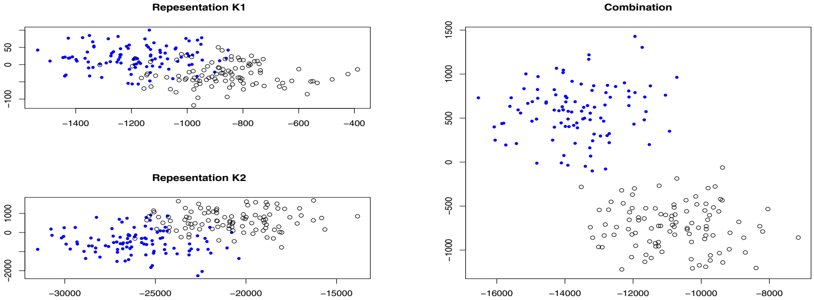
In our case, we obtain a Fusion Kernel for the matrix K^* assuming (following [10]) that every ϕ_h is defined by linear combinations of the eigenfunctions of $\tilde{K}_1, \ldots, \tilde{K}_p$. In practice, we do not know neither the eigenfunctions ϕ_h^* of the combined kernel matrix K_S^*, nor the eigenfunctions ϕ_{jr} of the Spectral kernels $\tilde{K}_{S1}, \ldots, \tilde{K}_{Sp}$. We only can compute $\hat{\gamma}_h$, the $h-th$ eigenvector of K_S^* and $\hat{\phi}_{jr}$ the jth eigenvector of the matrix K_{Sr}. However, the eigenfunctions of the kernel K^* can be approximated, up to a normalization factor, by the eigenvectors of the matrix K_S^* and the coefficients of the linear combinations of the eigenfunctions ϕ_{jr} approximating each γ_h can be approximated by a least squares projection of each $\hat{\gamma}_h$ onto the set of $\{\hat{\phi}_{jr}\}$ [10]. Finally, the eigenvalues λ_h of K^* in eq. (8) can be estimated using $\hat{\lambda}_h$, the eigenvalues of the matrix K_S^* (see [2,13] for details). However, if K_S^* is not positive definite, a transformation of them can be considered to guarantee the positive definiteness of the kernels. In this way we have all the ingredients for learning a kernel function corresponding to any kernel matrix obtained by combining the set of Spectral Kernels $\tilde{K}_{S1}, \ldots, \tilde{K}_{Sp}$.

4 Experiments

4.1 Kernels and Curves Projections: Illustrative Example

In this experiment we illustrate the behavior of our methodology in a simulated example. Consider two families of 4 dimensional curves sampled at 200 points: a) Class 1: $c(x) = \sum_{j=1}^{4} a_j \phi_j(x)$, where $a \sim N_4(\mu_1, \Sigma)$. b) Class 2: $c(x) = \sum_{j=1}^{4} b_j \phi_j(x)$, where $b \sim N_4(\mu_2, \Sigma)$ with $x \in [-5, 5]$, $\mu_1 = (2, 3, 3, 2)$, $\mu_2 = (2, 2, 2, 2)$, $\Sigma = diag(0.25, 0.25, 0.25, 0.25)$ and $\phi_1(x) = sin(x)$, $\phi_2(x) = cos(x)$, $\phi_3(x) = sin(2x)$, $\phi_4(x) = cos(2x)$. We generated 100 curves of each class. See Figure 1.

We consider two kernels to project the data onto two different RKHS via eq. (5): $K_1(\mathbf{x}, \mathbf{y}) = 0.5(\mathbf{x}^T\mathbf{y}) + 1$ and K_2 the data covariance matrix. We project the curves for both kernels by solving problem 3 for each kernel using $\gamma = 0,0001$. We



(a) Projections of the curves onto the two first eigenfunctions of kernels K_1 (top) and K_2 (down).

(b) Projections of the curves onto the two first eigenfunctions of the kernel combination

Fig. 2. Curves projections by K_1, K_2 and the AKM method

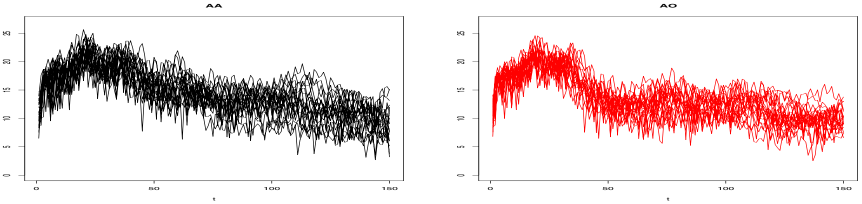


Fig. 3. Two classes of curves of the Phoneme data set

show the two first components of the projected curves in Figure 2 (a). It is apparent that none of the projections are able to separate the two classes of curves. Next we combine the Spectral kernels $\tilde{K}_1$ and $\tilde{K}_2$ resulting from both representations by using the Kernel Fusion with the MAKM procedure as combination method (see [5] for details). The projection onto the two first eigenfunctions are shown in Figure 2 (b). Now the two classes become (linearly) separable in the fusion space.

4.2 Phoneme Data Classification

The original data [6] set correspond to 2000 discretized log-periodograms of the phonemes "sh", "iy", "dcl", "aa" and "ao". Each phoneme is associated with a class of the experiment. We consider in this example those curves corresponding to the phonemes "aa" and "ao" since they present similar periodograms and are difficult to classify. A plot of 25 series of each class is shown in Figure 3.

We consider several RKHSs induced by Gaussian kernels $K_i(\mathbf{x}, \mathbf{y}) = \exp\{-\sigma_i \|\mathbf{x}-\mathbf{y}\|^2\}$ with a broad range of parameters $\sigma_i \in \{10, 7.5, 5, 2.5, 1, 0.1, 0.001\}$. We consider $\gamma = 0.001$ in eq. (3) and we project the curves using eq. (6). We estimate the Spectral kernels $\tilde{K}_i$ for $i = 1, \ldots, 7$ of the representations by using. (7) and we obtain the the Fusion Kernel in this case for the following combinations methods:

Table 1. Comparative results for the Phoneme Data after 100 runs

Method	Train Error.	Std. Dev.	Test Error	Std. Dev
Raw data	0.0682	(0.0039)	0.2606	(0.0097)
$RBF_{\sigma=10}$	0.1796	(0.0022)	0.2075	(0.0083)
$RBF_{\sigma=7.5}$	0.1787	(0.0029)	0.2037	(0.0069)
$RBF_{\sigma=5.0}$	0.1950	(0.0020)	0.2137	(0.0082)
$RBF_{\sigma=2.5}$	0.1975	(0.0019)	0.2162	(0.0068)
$RBF_{\sigma=1.0}$	0.2198	(0.0024)	0.2200	(0.0094)
$RBF_{\sigma=0.1}$	0.2242	(0.0030)	0.2243	(0.0105)
$RBF_{\sigma=0.001}$	0.2896	(0.0037)	0.2825	(0.0085)
$Fusion\ Kernel_{AKM}$	0.1868	(0.0033)	0.1906	(0.0080)
$Fusion\ Kernel_{MAKM_{\tau=1}}$	0.0000	(0.0000)	0.1881	(0.0092)
$Fusion\ Kernel_{MAXMIN}$	0.0000	(0.0000)	0.1862	(0.0069)
$Fusion\ Kernel_{AV_{\tau=1}}$	0.0000	(0.0000)	0.1875	(0.0079)
PSR	0.1866	(0.0085)	0.2033	(0.0028)
$NPCD_{derivative}$	0.2205	(0.0009)	0.3468	(0.0034)
$MPLSR_5$	0.1106	(0.0009)	0.1928	(0.0031)

AKM, MAKM, MAXMIN and AV. We use 80% of the data for training and 20% for testing and we then apply a SVM for classification (with penalization term $C = 100$) for the seven original Spectral kernels and the four Fusion Kernel combinations. We also use (for comparison purposes) two specific techniques designed to deal with functional data that have been shown to obtain very competitive results: P-spline signal regression (PSR) [8] and NPCD/MPLSR [6] with second derivative and PLS semimetrics (for dimensions 4,5,6,7 and 8). In addition, we include the results for a SVM (with linear kernel) on the raw data to compare classification results with a competitive technique that does not preprocess the data. Results are shown in Table 1.

Classifications errors in this example are large for any technique due to the overlapping of the curves. Regarding the initial RBF projections, all of them, excepting that corresponding to $\sigma = 0.001$, are able to improve the performance of a linear SVM in a particular favorable case for the linear SVM [9]. In particular, the best result for the five initial projection is given by $\sigma = 7.5$ with a 20.37% of misclassification data. However, this result is improved by the proposed fusion procedure (for all the combinations techniques) and also outperform the PSR and the MPLSR methods. Specially accurate are the results for the combinations that use labels in the fusion process as $MAKM$, $MAXMIN$ and AV that achieve errors of 18.81% and 18.62% and 18.75 respectively.

5 Conclusions

In this work we proposed a methodology for combining classifiers for functional data (time series in this paper). By considering different kernel functions we induce different SVM classifiers and then we combine them obtaining a true kernel function with the help of a Spectral Kernel called Fusion Kernel. The idea is represent functional data by means if the eigenfunctions of kernels resulted to be interesting from the theoretical and practical point of view. The experimental results show how this methodology significantly improves the existent procedures specifically designed for time series classification. It is quite remarkable that this point of view provides different sets of basis functions in a very natural way.

References

1. Aroszajn, N.: Theory of Reproducing Kernels. Transactions of the American Mathematical Society 68(3), 337–404 (1950)
2. Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.-F., Vincent, P., Ouimet, M.: Learning eigenfunctions links spectral embedding and kernel PCA. Neural Computation 16, 2197–2219 (2004)
3. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proc. Fifth ACM Workshop on Computational Learning Theory (COLT), pp. 144–152. ACM Press, New York (1992)
4. Cucker, F., Smale, S.: On the Mathematical Foundations of Learning. Bulletin of the American Mathematical Society 39(1), 1–49 (2002)
5. de Diego, I.M.n., Moguerza, J.M., Muñoz, A.: Combining Kernel Information for Support Vector Classification. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 102–111. Springer, Heidelberg (2004)
6. Ferraty, F., Vieu, P.: Curves discrimination: a nonparametric functional approach. Computational Statistics & Data Analysis 44, 161–173 (2003)
7. González, J., Muñoz, A.: Representing Functional Data using Support Vector Machines. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 332–339. Springer, Heidelberg (2008)
8. Marx, B., Eilers, P.: Generalized linear regression on sampled signals and curves: a P-spline approach. Technometrics 41(1), 1–13 (1999)
9. Moguerza, J.M., Muñoz, A.: Support Vector Machines with Applications. Statistical Science 21(3), 322–357 (2006)
10. Muñoz, A., González, J.: Functional Learning of Kernels for Information Fusion Purposes. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 277–283. Springer, Heidelberg (2008)
11. Parzen, E.: On Recent Advances in Time Series Modelling. IEEE Transactions on Automatic Control AC-19, 723–730 (1977)
12. Ramsay, J.O., Silverman, B.W.: Functional Data Analysis, 2nd edn. Springer, New York (2006)
13. Schlesinger, S.: Approximating Eigenvalues and Eigenfunctions of Symmetric Kernels. Journal of the Society for Industrial and Applied Mathematics 6(1), 1–14 (1957)
14. Schölkopf, B., Herbrich, R., Smola, A.J., Williamson, R.C.: A Generalized Representer Theorem. In: Helmbold, D.P., Williamson, B. (eds.) COLT 2001 and EuroCOLT 2001. LNCS (LNAI), vol. 2111, pp. 416–426. Springer, Heidelberg (2001)
15. Wahba, G.: Spline Models for Observational Data. Series in Applied Mathematics, vol. 59. SIAM, Philadelphia (1990)

Finding Small Consistent Subset for the Nearest Neighbor Classifier Based on Support Graphs

Milton García-Borroto[1,3], Yenny Villuendas-Rey[2], Jesús Ariel Carrasco-Ochoa[3], and José Fco. Martínez-Trinidad[3]

[1] Bioplantas Center, UNICA, Carretera a Morón km 9 ½, C. de Ávila, Cuba
mil@bioplantas.cu
http://www.bioplantas.cu
[2] Ciego de Ávila University UNICA, Carretera a Morón km 9 ½, C. de Ávila, Cuba
yennyv@bioplantas.cu
http://www.unica.cu
[3] National Institute of Astrophysics, Optics and Electronics, Puebla, México
{ariel,fmartine}@inaoep.mx
http://www.inaoep.mx

Abstract. Finding a minimal subset of objects that correctly classify the training set for the nearest neighbors classifier has been an active research area in Pattern Recognition and Machine Learning communities for decades. Although finding the Minimal Consistent Subset is not feasible in many real applications, several authors have proposed methods to find small consistent subsets. In this paper, we introduce a novel algorithm for this task, based on support graphs. Experiments over a wide range of repository databases show that our algorithm finds consistent subsets with lower cardinality than traditional methods.

Keywords: nearest neighbor, condensing, prototype selection, minimal consistent subset.

1 Introduction

Instance based learning has become one of the most used techniques in Machine Learning and Pattern Recognition. Among these techniques, the Nearest Neighbor (NN) classifier is one of the most popular supervised classifiers, due to its simplicity and high accuracy results. Two of its major advantages are that it does not assume any knowledge about data distribution, and its error is asymptotically bounded by twice the Bayes error [1].

However, in order to classify a new instance, NN computes its distance with all the objects in the training set. The computational cost of classification depends not only on the amount of objects in the training set, but also on the distance function, which in some domains, such as image processing, can be computationally expensive [2]. In addition, the storage cost depends on the amount of instances, and the number of features that describe them.

Since the introduction of the NN classifier, there is a constant research interest to find a reduced subset of instances with approximately the same classification power than the original set. Deleting redundant and irrelevant objects of the training set

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 465–472, 2009.

seems to be one of the most successfully approach. By using a reduced set of objects, or condensed subset, it is possible to decrease both storage and classification cost.

There are subsets especially important in object reduction, named consistent subsets, which are those subsets that correctly classify the whole training set. Between them, those with minimum size, are the focus of many researches and papers. Although finding a consistent subset of minimum cardinality is a NP-complete problem [3], finding a small subset is a common goal in many researches [4-6] and it is still a challenge.

In this paper, we introduce a new method for obtaining an approximation to the Minimal Consistent Subset (MCS) for the Nearest Neighbor classifier. It iteratively selects the less useful object for keeping the consistency using the information on a support graph. A support graph includes all the information about which objects guarantee the correct classification of other objects. This new method frequently obtains an object subset with fewer objects than previous methods reported in the literature.

The paper is organized as follows: section two summarizes some of the previous works to find minimal consistent subsets, section three introduces our proposal, section four details the numerical results of the experiments and section five offers some conclusions.

2 Minimal Consistent Subset Selection

Finding consistent subsets of objects for the Nearest Neighbor classifier has been a problem of interest in Pattern Recognition since 1968 when Hart proposed the Condensed Nearest Neighbor (CNN) algorithm [4]. In this work, he introduced the concept of consistent subset, a subset of the objects that correctly classifies all samples in the training set. In 1991, Wilfong demonstrated that finding a consistent subset of minimum cardinality is a NP-complete problem [3]. However, since Hart´s work, there has been several attempts to find small consistent subsets.

Among the most cited methods for finding minimal consistent subsets are RNN [5] and MCS [6]. The Reduced Nearest Neighbor (RNN) consists in a post processing of the CNN algorithm. After applying CNN, RNN deletes an object if this deletion does not introduce any inconsistency. Gates [5] demonstrated that if the minimum consistent subset is a subset of the CNN result, then RNN method always finds it.

In MCS, every instance x gives a vote to each instance (of the same class) closer than the Nearest Unlike Neighbor (NUN) object. The NUN is the object from different class closest to x. The MCS algorithm iteratively constructs a consistent subset, adding the instance with most votes, and repeating the process until a consistent subset is found.

The Generalized CNN [7] is another method that extract a consistent subset using a stronger criterion for removing objects. Due to this, it always obtains supersets of CNN result, with larger size.

3 Condensation Based on Support Graphs

In this section, we introduce the CSESupport algorithm. It is a modification of the CSE algorithm [8], aiming to obtain a better approximation to the minimal consistent subset. It has the following changes:

- CSESupport uses a support graph instead of a nearest neighbor graph. A support graph is a directed graph, having arcs $x \rightarrow y$ if and only if the object y supports the object x. We say that y supports x if y is closer to x than x's NUN. This way if y belongs to the condensed subset, it guarantees the correct classification of x with a NN classifier. In a support graph, such vertexes with more inward arcs are the most important for condensing, because they support more objects.
- CSESupport makes multiple iterations. After each iteration, the algorithm extracts a smaller consistent subset than previous step, so the NUNs of some objects can differ. This way, the support graph can change, leading to potential new reductions.

Each iteration of CSESupport contains the following steps:

1. Support graph construction. The graph is constructed with respect to the objects in the current solution.
2. Add nodes with no successors (so they can not be supported by any other object) to the new solution if they are not assured, using the procedure *Move*. The *Move* procedure has three steps:
   a. Mark all antecessors of the node as assured. An assured node can be safely discarded from the training sample without affecting consistency.
   b. Delete all shared non-simultaneous deletion marks. These marks are a key point inherited from CSE to avoid inconsistencies in the result. They are added to the objects that support a deleted object, and if an object is the last one with such mark, it has to be included in the result. It is clear than the marks used by an object are different by those used by other object.
   c. Delete the node from the graph. Note than the assured nodes are not discarded in this step, because their inclusion in the result can assure more objects. This frequently led to smaller results.
3. Add nodes with the last non-simultaneous deletion mark to the new solution using Move.
4. Delete the less important node in the graph using the Discard procedure. A node is considered less important than other if it contains less inward arcs, because it is able to support less objects. Including in the result an object that supports more objects that other usually leads to smaller size consistent subsets. This criterion guides the CSESupport heuristic.
   The *Discard* procedure has three steps:
   a. If the node is not already assured, mark all its successors with a non-simultaneous deletion marks.
   b. Delete x of every non-simultaneous deletion marks in which it appears.
   c. Delete the node from the graph.
5. If the graph is not empty, go to step 2.

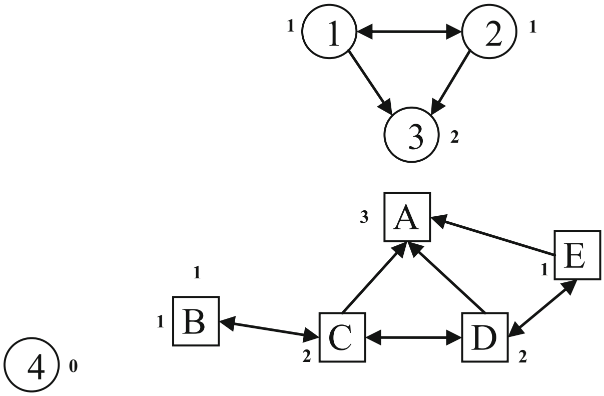


Fig. 1. Support graph with 9 object in 2 classes: circles and squares. The small numbers close to the shapes contains the number of inward arcs.

After an iteration the algorithm stops if the current solution is equal than previous solution. Otherwise, a new iteration is executed using the current solution to build the new support graph.

Now we will run an iteration of CSESupport with the objects in Fig. 1. In this example we build the support graph using the Euclidean distance between the centers of the shapes.

In step 2 nodes 3, 4 and A are *moved* to the current solution, because they have no successors. While *moving*, nodes 1, 2, C, D and E are assured. In step 4 the node 1 is *deleted*, but like it is assured no marking is necessary. In step 5 we loop to step 2. Step 2 can not be applied because the only node with no support (object 2) is already assured. In step 4 object 2 is *deleted.* Similarly object E is deleted in the next loop. Next loop node B is deleted in step 4, but like it is not assured the node C is marked with a non-simultaneous deletion mark. Next loop the step 3 *moves* node C to the result. Finally objects C and D are *deleted* without marking because they were previously assured. The resultant consistent subset is then {3,4,A,C}.

CSESupport, like CSE, is not able to deal with general k-NN classifiers, because the support graph only contains the information about the nearest neighbor.

4 Experimental Results

In order to test the behavior of CSESupport for finding small consistent subsets, we select RNN and MCS methods, which have been commonly used in experimental comparisons. Other methods, like CNN and GCNN, always produce larger results than RNN, so we dismiss them. For comparing the performance of the selected condensing methods, first we applied them over 2-D synthetic databases (section 4.1). We also carried out numerical experiments over a wide-range of databases from the UCI repository of Machine Learning [9] (section 4.2). The description of the repository databases appears in Tables 1.

The experiments with repository databases use 10-fold cross validation. Since NN classifier results depend on the function used for comparing objects, we use two functions: HEOM and HVDM [10]. In order to determine the best method, we compute object retention for each one of them, and count how many times each method had the lower retention rate. For determining the reduction influence in classifier accuracy, we made a two-tailed T-Test with significance of 0.05, with respect to the method with lower classifier error, and compute how many times each method had the lower error, according to the T-Test.

Table 1. Description of repository databases

Database	non-num/ num feats	Objects	Database	non-num/ num feats	Objects
anneal	29/9	798	**hepatitis**	13/6	155
autos	10/16	205	**iris**	0/4	150
balance-scale	0/4	625	**labor**	6/8	57
breast-cancer	9/0	286	**lymph**	15/3	148
breast-w	0/9	299	**post-pat-data**	7/1	90
cmc	7/2	1473	**sonar**	0/60	208
colic	15/7	368	**tae**	2/3	151
credit-a	9/6	690	**trains**	29/4	10
cylinder-bands	20/20	512	**vehicle**	0/18	846
dermatology	1/33	366	**vote**	16/0	435
glass	0/10	214	**vowel**	3/9	990
heart-c	7/6	303	**zoo**	16/1	101
heart-h	7/6	294			

4.1 Results with Synthetic Databases

We generate two 2-D synthetic databases (see Fig. 1). Each database has three classes, represented by triangles, squares and circles, respectively. Database "Bananas and circle" is a modified version of the database "Bananas", used by Kuncheva [11]. Database "Venn´s diagram" has three overlapped classes, forming circles as in a Venn diagram. In gray, silver gray and white we represent the decision region of the classes squares, triangles and circles, respectively.

In figures 2-3, we show the results of each method over synthetic databases, using the Euclidean distance. The CSESupport method outperforms RNN in one and MCS in both synthetic databases.

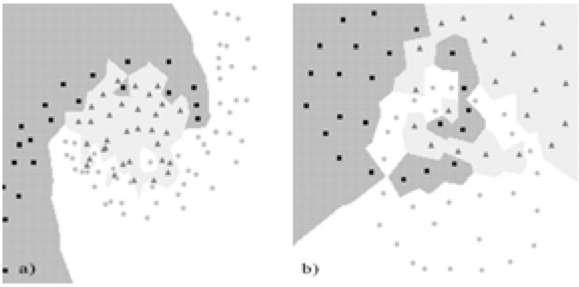


Fig. 2. a) "Bananas and circle", b) "Venn´s diagram"

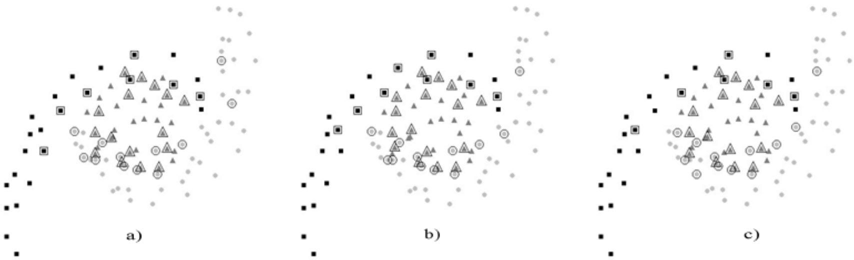


Fig. 3. Results of each method in database “Bananas and circle”. a) RNN (35 obj.), b) MCS (36 obj.) and c) CSESupport (33 obj.).

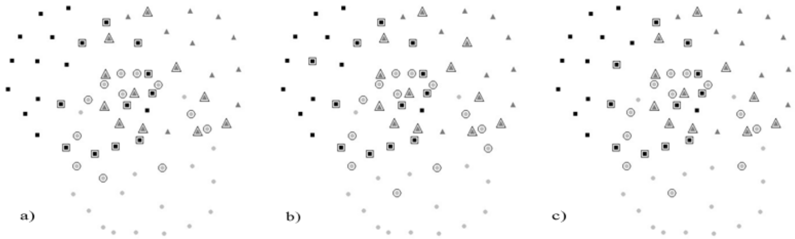
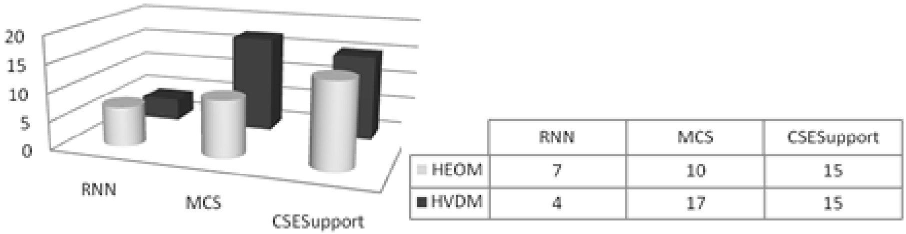


Fig. 4. Results of each method in database “Venn´s diagram”. a) RNN (34 obj.), b) MCS (37 obj.) and c) CSESupport (34 obj.).

4.2 Results with Repository Databases

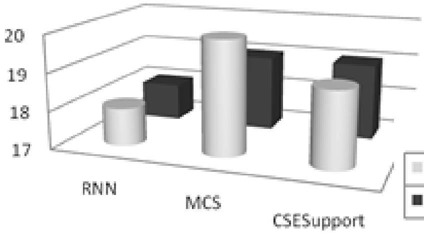
For repository databases, we averaged for each dissimilarity function the object retention results and counted how many times each method had the lowest object retention (see Fig. 5). According to our experiments, CSESupport achieved the best results, obtaining consistent subsets of lower cardinality more often than RNN and MCS.

These results can be explained because the quality of RNN result is strongly dependant on the CNN result. The CNN method is, on the other hand, strongly dependant on the order of the objects in the database. With respect to MCS, the difference is mainly due to the general search strategy: MCS inserts the most useful object, while CSESupport deletes the less important object. This way, like in other pattern recognition problems [12], sequential backward searches are more accurate than sequential forward searches.



	RNN	MCS	CSESupport
▨ HEOM	7	10	15
■ HVDM	4	17	15

Fig. 5. Number of times each method achieved the lowest object retention



	RNN	MCS	CSESupport
□ HEOM	18	20	19
■ HVDM	18	19	19

Fig. 6. Number of times each method achieved the lowest classification error

We also compared the performance of the methods according to classification error. We made a two-tailed T-Test [13] with respect to the lowest error result, including the classifier trained with the original training set. We also count how many times each method had (statistically) the lowest error. The summary of the error classification results appears in Fig. 6. In general, all methods had similar results, maintaining low classification error in most of the tested databases.

5 Conclusions

Finding the minimal consistent subset is a computational intractable problem in many real life databases. Nevertheless, some authors have introduced heuristic proposals to find small consistent subsets. In this paper, we introduce a new algorithm for finding low cardinality consistent subsets for Nearest Neighbor classifiers, which usually find smaller subsets than state-of-the-art methods. The proposed method is based on support graphs, which provides valuable information in regions where objects from different classes are close together. In our experiments, the proposed condensing method (CSESupport) achieved the best results according to object retention, obtaining subsets with the lowest cardinality more often than MCS and RNN. As future work, we are working on including more information in the support graph to make CSESupport able to deal with k-NN classifiers.

Acknowledgments

The first author wants to thank to the National Institute of Astrophysic, Optics and Electronics for its support on this research. This work is partly supported by National Council of Science and Technology of México under grant 252752.

References

1. Cover, T., Hart, P.E.: Nearest Neighbor pattern classification. IEEE Trans. on Information Theory 13, 21–27 (1967)
2. Athitsos, V.: Learning embeddings for indexing, retrieval, and classification, with applications to object and shape recognition in image databases. Vol. Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, p. 156. Boston University (2006)

3. Wilfong, G.: Nearest neighbor problems. In: 7th Annual ACM Symposium on Computational Geometry, pp. 224–233 (1991)
4. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. on Information Theory 14, 515–516 (1968)
5. Gates, G.W.: The reduced nearest neighbor rule. IEEE Transactions on Information Theory IT-18, 431–433 (1972)
6. Dasarathy, B.D.: Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Transactions on Systems, Man and Cybernetics 24, 511–517 (1994)
7. Chou, C.-H., Kuo, B.-H., Chang, F.: The Generalized Condensed Nearest Neighbor Rule as a Data Reduction Method. In: 18th International Conference on Pattern Recognition ICPR 2006, Tampa, USA. IEEE, Los Alamitos (2006)
8. García-Borroto, M., Ruiz-Shulcloper, J.: Selecting Prototypes in Mixed Incomplete Data. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 450–459. Springer, Heidelberg (2005)
9. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases. University of California at Irvine, Department of Information and Computer Science, Irvine (1998)
10. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
11. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. Wiley-Interscience, Hoboken (2004)
12. Pudil, P., Novovicova, F.J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. 15, 1119–1125 (1993)
13. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, vol. 10, pp. 1895–1923. MIT Press, Cambridge (1998)

Analysis of the GRNs Inference by Using Tsallis Entropy and a Feature Selection Approach

Fabrício M. Lopes[1,2], Evaldo A. de Oliveira[1], and Roberto M. Cesar-Jr[1]

[1] Institute of Mathematics and Statistics, University of São Paulo, Brazil
{fabriciolopes,evaldo,cesar}@vision.ime.usp.br
[2] Federal University of Technology - Paraná, Brazil
fabricio@utfpr.edu.br

Abstract. An important problem in the bioinformatics field is to understand how genes are regulated and interact through gene networks. This knowledge can be helpful for many applications, such as disease treatment design and drugs creation purposes. For this reason, it is very important to uncover the functional relationship among genes and then to construct the gene regulatory network (GRN) from temporal expression data. However, this task usually involves data with a large number of variables and small number of observations. In this way, there is a strong motivation to use pattern recognition and dimensionality reduction approaches. In particular, feature selection is specially important in order to select the most important predictor genes that can explain some phenomena associated with the target genes. This work presents a first study about the sensibility of entropy methods regarding the entropy functional form, applied to the problem of topology recovery of GRNs. The generalized entropy proposed by Tsallis is used to study this sensibility. The inference process is based on a feature selection approach, which is applied to simulated temporal expression data generated by an artificial gene network (AGN) model. The inferred GRNs are validated in terms of global network measures. Some interesting conclusions can be drawn from the experimental results, as reported for the first time in the present paper.

Keywords: Tsallis entropy, feature selection, inference, validation, gene regulatory networks, bioinformatics.

1 Introduction

In general, living organisms can be viewed as networks of molecules connected by chemical reactions. More specifically, the cell control involves the activity of several related genes, in which the relationship among them is known or not. Gene regulatory networks (GRNs) are used to indicate the interrelation among genes in the genomic level [1]. Such information is very important for disease treatment design, drugs creation purposes and to understand the activity of living organisms in the molecular level. In this way, there is a strong motivation for GRNs inference.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 473–480, 2009.

High-throughput techniques for measurement of mRNA concentrations allow the simultaneous verification of cell components activity (state) in multiple instants of time. Some methods were proposed for modeling and identification of GRNs from expression data sets [2,3,4,5,6,7,8,9,10,11]. This work focuses attention in a particular method proposed by Barrera *et al* [7] in order to evaluate the application of Tsallis entropy [12] in the GRNs inference problem by using an artificial gene network (AGN) model [13]. This technique is based on a feature selection algorithm that minimizes entropy measures as the criterion function.

The Tsallis entropy has been stood out in the last years as a generalization of the Shannon entropy [14]. This is not only due to its applications [15], but also due to its theoretical foundation [16]. Its use becomes important on systems with long-range interactions (which causes long-range correlations), a particular feature of nonextensive systems. But, are the gene regulatory networks nonextensive? How to interpret them in this context? In order to investigate these questions, the present work addresses the problem about the inference and extensivity of GRNs by applying information theory. We also analyze the quality and limitations of the adopted method [7] to infer network topologies.

Next section presents a brief description of the AGN model to generate the ground-truth and the simulated expression signals. Section 3 presents the network inference method, while Section 4 describes the similarity measures adopted to compare the inferred and the ground-truth networks. Experimental results are presented and discussed in Section 5. Section 6 concludes this text with possible future directions of this work.

2 AGN Model

The AGN model [13] is composed of three main components: (1) topology, (2) network state and (3) transition functions. The topology of an AGN may be defined by theoretical complex networks models [17,18,19]. We have adopted the uniformly-random Erdös-Rényi (ER) and the scale-free Barabási-Albert (BA) models.

The AGN model is a complex network $G = (V, E)$ formed by a set $V = \{v_1, v_2, \ldots, v_N\}$ of nodes or "genes", connected by a set $E = \{e_1, e_2, \ldots, e_M\}$. It is important to keep the same average number of connections of nodes k for comparative analysis between ER and BA. In this way, in order to keep k fixed for the ER model, the probability p of connecting each pair of nodes is given by $p = \frac{k}{N-1}$. The BA topology follows a *linear preferential attachment* rule, *i.e.*, the probability of the new node v_i to connect to an existing node v_j is proportional to the degree of v_j. Therefore, the nodes of ER networks have a random pattern of connections while BA networks have some nodes highly connected and others with few connections.

Each gene can assume a value from a discrete set D (in this work, $D = \{0, 1\}$, *i.e.*, on/off) that represents its states. The network state s at time t is determined by $s_t = \{v_{1,t}, v_{2,t}, \ldots, v_{N,t}\}$, called the state vector.

The transition functions F are defined by logic circuits, one for each gene of the network $v_{i,t+1} = F(u_{ki,t})$, in which $u_{ki} \in G$ represents the k genes (predictors) that have input edges to v_i (target). The number of predictors and its influence (measured by edges) on target genes are defined by considering a deterministic model [20], *i.e.*, the networks remain fixed in the choice of k input nodes, chosen logic circuits and chosen predictors, during all instants of time.

The dynamics of an AGN is determined by applying the transition functions to an arbitrary initial state $s_1 = \{v_1 = 1, v_2 = 0, \ldots, v_N = 1\}$ during T time instants (*i.e.*, the signal size). The target state at time $t_{i+1}, i = 2, 3, \ldots, T$ is hence obtained by observing the predictor states at t_i and by applying its respective logic circuit. As a result, we have the simulated temporal signals of length T, which are used for the network identification method presented in Section 3.

3 Network Inference

The use of entropy functions to infer gene interaction network topology from time series has been showed a promising tool [7,21]. Of course, the precision of the inference depends on the information available and the suitability of its use.

The inference method used in this work is described in [7], in which the entropy [14] of the temporal gene expression was employed as a criterion function in a feature selection [22] approach to identify the network topology. The main idea is to select the predictors subset that minimizes the entropy of each target gene from its expressions profiles.

In this context, network inference is modeled as a series of feature selection problems, one for each gene. The inference method starts by fixing the target gene Y. The time series determined by gene expressions are used to build a table of conditional probabilities of the classes Y given the patterns $\mathbf{X}$ that minimizes a criterion function. The classes are defined by the target values at time $t + 1$, while the patterns are defined by the predictors values at time t.

The search space is normally very large, so that an exhaustive search cannot be performed. In our approach, the *Sequential Forward Floating Search* (SFFS) [23] algorithm was applied for each target gene in order to select the subset of genes $\mathbf{X}$ that minimizes the criterion function given by Equation (1). As defined in [24], the penalized mean conditional entropy of Y given all the possible instances $\mathbf{x} \in \mathbf{X}$ is given by:

$$H(Y \mid \mathbf{X}) = \frac{\alpha(M - N)\ H(Y)}{\alpha M + s} + \frac{\sum_{i=1}^{N}(f_i + \alpha)\ H(Y \mid \mathbf{X} = \mathbf{x_i})}{\alpha M + s}, \qquad (1)$$

where M is the number of possible instances of the feature vector $\mathbf{X}$, N is the number of observed instances (the number of non-observed instances is given by $M - N$), f_i is the absolute frequency (number of observations) of $\mathbf{x_i}$ and s is the number of samples. The α constant is the penalty weight for non-observed instances ($\alpha = 1$ in the present work).

Once we are interested to better understand the method, mainly about its performance given a data set, we focus on the entropy function and use the generalized entropy proposed by Tsallis [12,25]. The Tsallis entropy has been studied and applied by many researchers in different approaches (information theory [16], thermodynamics [26]) and systems. Its suitability has been proved [27], mainly where the Shannon is not recommended, *i.e.*, for long-range interactions between the nodes. As defined in [24], by the inclusion of such generalization in Equation (1), the conditional entropy $H(Y \mid \mathbf{X} = \mathbf{x_i})$ becomes:

$$H(Y \mid \mathbf{X} = \mathbf{x_i}) = \frac{1}{q-1}(1 - \sum_{y \in Y}(P(y \mid \mathbf{x_i}))^q), \tag{2}$$

where $P(y \mid \mathbf{x_i})$ is the conditional probability of y given the observation of an instance $\mathbf{x_i} \in \mathbf{X}$.

The Tsallis entropy generalizes the Shannon entropy and can be used as a functional set by the parameter q which is defined as an entropic parameter that characterizes the degree of nonextensivity. For $q < 1$ the entropies are superextensives and for $q > 1$ the entropies are subextensives. Furthermore, when $q = 1$ the entropies are extensive and the Shannon form is completely recovered[1].

Lower values of H yield better feature subspaces (the lower H, the larger is the information gained about Y by observing $\mathbf{X}$). In this way, the SFFS is guided to minimize the criterion function in Equation (1). The selected features are taken as predictor genes for each target gene, which are used to link the genes and thus to recover the network topology.

The next section describes the similarity measures adopted to compare the inferred and the AGN-based networks.

4 Validation

In order to quantify the similarity between A (AGN model networks) and B (inferred networks), we adopted the validation metric based on a confusion matrix [28] (Table 1).

Table 1. Confusion matrix. TP=true positive, FN=false negative, FP=false positive, TN=true negative.

Edge	**Inferred in B**	**Not Inferred in B**
Present in A	TP	FN
Absent in A	FP	TN

The networks are represented in terms of their respective adjacency matrices M, such that each edge from node i to node j implies $M(i,j) = 1$, with $M(i,j) = 0$ otherwise. The measures considered in this work are calculated as follows:

$$\begin{gathered} Similarity(A,B) = \sqrt{TPR \cdot TNR}\,, \\ TPR = \frac{TP}{(TP+FN)}, \quad TNR = \frac{TN}{(TN+FP)}. \end{gathered} \tag{3}$$

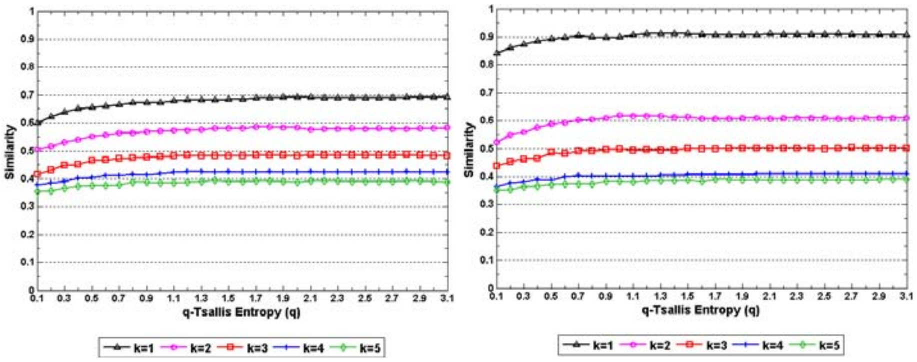
[1] This can be easily obtained by taking the limit $q \to 1$ in the Equation (2).

We consider the geometrical average $Similarity(A, B)$ between the ratios of correct ones (TPR) and correct zeros (TNR), observing the ground-truth matrix A and the inferred matrix B. Observe that both coincidences and differences are taken into account by these indices, implying the maximum similarity to be obtained for values near 1.

5 Experimental Results

This section presents the experimental results by applying Tsallis entropy [25] for GRNs inference, as presented in Section 3, by considering three main aspects: (1) Variation of parameter q of Tsallis entropy; (2) two different complex network topologies (ER) and (BA); (3) complexity of networks in terms of average node degree (k).

For all experiments, the two network models (BA and ER) with 100 nodes were used. The q parameter of Tsallis entropy varied from 0.1 to 3.1 in steps of 0.1, and the average node degree k varied from 1 to 5 in steps of 1. The simulated temporal expression was generated using 10 randomly chosen initial states, each one with length 30. These expressions were concatenated into a single signal of size 300. The experimental results were obtained from 50 simulations for each network topology and k value, using the default parameters of the method [24].



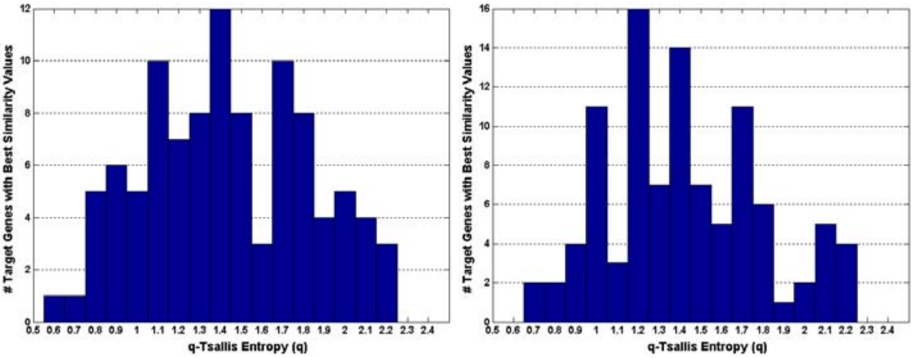
(a) ER

(b) BA

Fig. 1. Network identification rate by increasing q of the Tsallis entropy, using: (a) uniformly-random Erdös-Rényi (ER) and (b) scale-free Barabási-Albert (BA)

Figures 1 (a) and (b) show the median values of similarity (described in Section 4) between the inferred networks and AGN-based networks in terms of q of the Tsallis entropy and the average node degrees (k). These figures present a soft increase of similarity rate by increasing the q for all average degrees k. This result suggests a dependence of the method accuracy on the parameter q.

In order to better investigate this behavior, Figures 2 (a) and (b) present the histograms of the frequency of target genes with best similarity rate found for each q value, by considering respectively, ER and BA network topologies.



(a) ER

(b) BA

Fig. 2. Histogram of the frequency of target genes with best similarity value per q, over all average node degree k., using: (a) uniformly-random Erdös-Rényi (ER) and (b) scale-free Barabási-Albert (BA). All average node degree k was considered.

Table 2. The best results found for $q = 1$ and for all q values (global) by considering: (a) uniformly-random Erdös-Rényi network topology (ER) and (b) scale-free Barabási-Albert network topology (BA)

(a) ER

q	k	**TP**	**FP**	**TN**	**FN**
1	1	175	195	9630	0
global		179	121	9700	0
1	2	224	137	9639	0
global		228	62	9710	0
1	3	231	136	9633	0
global		241	71	9688	0
1	4	208	184	9608	0
global		218	106	9676	0
1	5	205	206	9578	11
global		210	139	9643	8

(b) BA

q	k	**TP**	**FP**	**TN**	**FN**
1	1	114	257	9629	0
global		114	193	9693	0
1	2	206	102	9692	0
global		206	27	9767	0
1	3	283	77	9636	4
global		290	20	9689	1
1	4	250	130	9508	112
global		289	78	9548	85
1	5	255	135	9423	187
global		283	105	9451	161

Figure 2 (a) presents higher frequency when $q = 1.4$, and the distribution is concentrated between $q = 0.6$ and $q = 2.2$. On the other hand, Figure 2 (b) presents higher frequency when $q = 1.2$, and the distribution is concentrated between $q = 0.7$ and $q = 2.2$. These results reinforce the fact that the variation of q is important for the method accuracy, *i.e.*, the nonextensivity of the networks.

In order to estimate the improvement of the accuracy by varying q, Tables 2 (a) and (b) present the comparisons of commonly used Shannon entropy $q = 1$ and the best global results obtained by the variation of q. In general, it is possible to notice that global results exhibit an improvement of accuracy *w.r.t* $q = 1$ for all average node degrees (k) in both network topologies (ER and BA). In particular, the number of false positives (FP) presents higher improvement of accuracy, achieving 55% of reduction in false positives for ER when $k = 2$ and 74% for BA when $2 \leqslant k \leqslant 3$.

6 Conclusion

This work described a comparative analysis in order to evaluate the application of Tsallis entropy in a GRN inference method based on a feature selection approach by considering three main aspects: (1) variation of the parameter q (degree of nonextensivity) of Tsallis entropy; (2) two different complex network topologies; (3) complexity of networks in terms of average node degree (k).

The results indicated a valuable improvement of the accuracy of the GRNs inference by using the Tsallis entropy. This improvement was observed in both kinds of networks (ER and BA) and also for different degrees of complexities k (average gene degree). We have found the best similarity values on the range $0.6 \leqslant q \leqslant 2.2$, where the degree of nonextensivity q around 1.4 performs better results. In fact, the results have shown that tested networks tend to be a little subextensive ($q > 1$).

These results can be seen as a first stage to better understand the inference of network topologies by information theory approaches, *i.e.*, by using entropy criteria. The main point is the possibility of nonextensivity of the networks and, therefore, the entropy related methods dependence on that.

The next stage of this work is to consider complex networks models and measurements [19] (local and global) and more precise similarity measures between two networks [29] in order to refine the analysis of the inference method and the nonextensivity of the networks. Other relevant improvement is to include some uncertainty in the transition functions, *i.e.*, to use stochastic transition functions and to measure its effect on network inference method. Other methods that apply information theory for GRNs inference could be included in the comparative results and analysis of nonextensivity of the networks.

Acknowledgments

This work was supported by FAPESP, CNPq and CAPES.

References

1. Hovatta, I., et al.: DNA microarray data analysis, 2nd edn. CSC, Scientific Computing Ltd. (2005)
2. Liang, S., Fuhrman, S., Somogyi, R.: Reveal: a general reverse engineering algorithm for inference of genetic network architectures. In: PSB, pp. 18–29 (1998)
3. Weaver, D.C., et al.: Modeling regulatory networks with weight matrices. In: PSB, pp. 112–123 (1999)
4. Butte, A., Kohane, I.: Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In: PSB, pp. 418–429 (2000)
5. Keles, S., van-der Laan, M., Eisen, M.B.: Identification of regulatory elements using a feature selection method. Bioinformatics 18(9), 1167–1175 (2002)
6. Shmulevich, I., et al.: Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18(2), 261–274 (2002)
7. Barrera, J., et al.: 2. In: Constructing probabilistic genetic networks of Plasmodium falciparum from dynamical expression signals of the intraerythrocytic development cycle, pp. 11–26. Springer, Heidelberg (2007)

8. Margolin, A.A., et al.: Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(suppl. 1) (2006)
9. Faith, J., et al.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biology 5(1), 259–265 (2007)
10. Meyer, P.E., Kontos, K., Lafitte, F., Bontempi, G.: Information-theoretic inference of large transcriptional regulatory networks. EURASIP Journal on Bioinformatics and Systems Biology, 1–9 (2007)
11. Zhao, W., Serpedin, E., Dougherty, E.R.: Inferring connectivity of genetic regulatory networks using information-theoretic criteria. IEEE/ACM Trans. on Comput. Biology and Bioinformatics 5(2), 262–274 (2008)
12. Tsallis, C.: Possible generalization of Boltzmann-Gibbs statistics. Journal of Statistical Physics 52(1), 479–487 (1988)
13. Lopes, F.M., Cesar-Jr, R.M., Costa, L.F.: AGN simulation and validation model. In: Bazzan, A.L.C., Craven, M., Martins, N.F. (eds.) BSB 2008. LNCS (LNBI), vol. 5167, pp. 169–173. Springer, Heidelberg (2008)
14. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
15. Issue, S.: Nonextensive statistical mechanics: new trends, new perspectives. Europhysics News 36(6), 185–231 (2005)
16. Abe, S.: Tsallis entropy: how unique? Cont. Mech. Therm. 16(3), 237–244 (2004)
17. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74(1), 47–97 (2002)
18. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 167–256 (2003)
19. Costa, L.F., Rodrigues, F.A., Travieso, G., Boas, P.R.V.: Characterization of complex networks: a survey of measurements. Adv. in Phys. 56(1), 167–242 (2007)
20. Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed nets. Journal of Theoretical Biology 22(3), 437–467 (1969)
21. Lopes, F.M., Martins-Jr, D.C., Cesar-Jr, R.M.: Comparative study of GRN's inference methods based on feature selection by mutual information. In: GENSIPS, pp. 1–4. IEEE Computer Society, Los Alamitos (2009)
22. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2000)
23. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recogn. Lett. 15(11), 1119–1125 (1994)
24. Lopes, F.M., Martins-Jr, D.C., Cesar-Jr, R.M.: Feature selection environment for genomic applications. BMC Bioinformatics 9(451), 1–21 (2008)
25. Tsallis, C.: Nonextensive Statistical Mechanics and its Applications. Lecture Notes in Physics. Springer, Heidelberg (2001)
26. Velazquez, L., Guzmán, F.: Remarks about the Tsallis formalism. Phys. Rev. E 65(4), 046134.1–046134.5 (2002)
27. Tsallis, C.: Nonadditive entropy: the concept and its use. The European Physical Journal A 40(3), 257–266 (2009)
28. Costa, L.F., et al.: Predicting the connectivity of primate cortical networks from topological and spatial node properties. BMC Systems Biology 1, 1–16 (2007)
29. Dougherty, E.R.: Validation of inference procedures for gene regulatory networks. Current Genomics 8(6), 351–359 (2007)

Clustering Ensemble Method for Heterogeneous Partitions

Sandro Vega-Pons and José Ruiz-Shulcloper

Advanced Technologies Application Center (CENATAV), Havana, Cuba
{svega,jshulcloper}@cenatav.co.cu

Abstract. Cluster ensemble is a promising technique for improving the clustering results. An alternative to generate the cluster ensemble is to use different representations of the data and different similarity measures between objects. This way, it is produced a cluster ensemble conformed by heterogeneous partitions obtained with different point of views of the faced problem. This diversity enhances the cluster ensemble but, it restricts the combination process since it makes difficult the use of the original data. In this paper, in order to solve these limitations, we propose a unified representation of the objects taking into account the whole information in the cluster ensemble. This representation allows working with the original data of the problem regardless of the used generation mechanism. Also, this new representation is embedded in the WKF [1] algorithm making a more robust cluster ensemble method. Experimental results with numerical, categorical and mixed datasets show the accuracy of the proposed method.

Keywords: Cluster ensemble, object representation, similarity measure, co-association matrix.

1 Introduction

Cluster ensemble has emerged as a good alternative to improve the quality of clustering results. Traditionally, given a set of objects, a cluster ensemble method consists in two principal steps: Generation, which is about the conformation of a set of partitions of these objects, and Consensus Function, where a new partition which is the *integration* of all partitions obtained in the generation step, is computed.

In the generation step, different representations of the objects can be used or the same representation with different similarity (dissimilarity) measures to obtain each partition in the cluster ensemble. Then, if it is necessary to work with the original data after the generation step, we have to deal with the following questions: Which representation of the objects should be used? Which similarity (dissimilarity) measure should be applied?

To the best of the authors knowledge, these questions have not been boarded before. Taking into account these situations, and giving them an adequate treatment, we can improve the quality of the clustering ensemble algorithms and

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 481–488, 2009.

enlarge their scope. Then, the main goal of this paper is to give an answer to these questions. In order to do that, we present a new way of representing the objects unifying the information in all partitions in the cluster ensemble. This representation is based on a new similarity matrix, which is obtained from the co-association of the objects in the clusters of the set of partitions, but taking into account more information than the traditional co-association matrix [2] and therefore expressing better the relationship between objects. By using this new representation of the objects can be applied, for example, any distance function and compute centroids in this new representation space, even when the original data are mixed (composed by numerical and non-numerical attributes).

This paper is organized as follows: In section 2 some related works are presented, highlighting the method proposed in [1] called WKF, as well as the motivation and problem description are outlined. Section 3 introduces our proposal and presents a modification of the WKF method with our new object representation embedded in the steps of this method. Experimental results are discussed in Section 4 and finally in Section 5 are the conclusions of our research.

2 Problem Discussion

We will denote $X = \{x_1, x_2, \ldots, x_n\}$ the set of objects, where each object is a tuple of some $d-$dimensional feature space Ω^d. $\mathbb{P}=\{P_1, P_2, \ldots, P_m\}$ is a set of partitions, where each $P_i = \left\{C_i^1,\ C_i^2, \ldots,\ C_i^{k_i}\right\}$ is a partition of the set of objects X, and C_i^j is the j^{th} cluster of the i^{th} partition, for all $i = 1, \ldots, m$. The goal of clustering ensemble methods is to find a partition P^*, which better represents the properties of each partition in $\mathbb{P}$.

Several clustering ensemble methods have been proposed in the literature, e.g. Co-association methods [2] and Hyper-Graph methods [3]. In these methods, it is not necessary to work with the original data after the generation step, i.e., once the set of partitions $\mathbb{P}$ is obtained, all the operations to obtain the consensus partition P^* are achieved taking into account only the partitions in $\mathbb{P}$.

For example, the co-association methods firstly compute the co-association matrix $\mathcal{C}$, where each cell has the following value:

$$\mathcal{C}_{ij} = \frac{1}{m}\sum_{k=1}^{m} \delta\left(P_k\left(x_i\right), P_k(x_j)\right) \tag{1}$$

$P_k\left(x_i\right)$ represents the associate label of the object x_i in the partition P_k, and $\delta\left(a, b\right)$ is 1, if $a = b$, and 0 otherwise. Then, the value in each position $\left(i, j\right)$ of this matrix is a measure about how many times the objects x_i and x_j are in the same cluster for all partitions in $\mathbb{P}$. Using the co-association matrix $\mathcal{C}$ as the similarity measure between objects, the consensus partition is obtained by applying a clustering algorithm.

The Hyper-graph methods start by representing each partition in the cluster ensemble with a hyper-edge. Then, the problem is reduced into a hyper-graph partitioning problem. Three efficient heuristics CSPA, HGPA and MLCA are presented in [3].

However, the set of objects X and their similarity (dissimilarity) values are additional information that, if it is well-used, the combination results can be improved. In other words, more complex methods that make use of this information in order to achieve better results can be developed. This is the case of the WKF method [1].

The WKF method uses the set of objects X and their similarities in an intermediate step, between the generation and the consensus function called Qualitative Analysis of the Cluster Ensemble (QACE). In this new step, it is estimated the relevance of each partition for the cluster ensemble.

The idea is to assign a weight to each partition that represents how relevant it is in the cluster ensemble. In this step, partitions which represent noise for the cluster ensemble are detected, and its effect in the consensus partition is minimized. The impact of this step in the final consensus partition is meaningful, since by using the information obtained in this step, a fairer combination process is achieved. In [1] the QACE step is performed by applying different cluster validity indexes, where each one of them measures the fulfillment of a particular property, e.g., compactness, separability, connectivity. Thus, to a partition that behaves very different to the rest of the partitions with respect to this properties, it is assigned a small weight, because it is probably a noisy partition obtained by a not appropriate generation mechanism. Otherwise, if a partition has an average behavior, it will have a higher weight assigned.

The consensus partition in the WKF method is computed by using a consensus function able to work with the weights computed in the QACE step. For this, each partition is represented by a graph. Furthermore, to obtain an exact representation of the consensus into a Hilbert Space, a kernel function is used as the similarity measure. Finally, an efficient stochastic search strategy is applied to obtain the final consensus partition.

Despite of the advantages that the QACE step offers, it has the limitation that the similarity between the objects must be computed on the original data X. This may cause the appearance of some problems such as:

1. The partitions could be created by using different representations of the data, either by completely different representations given by different modelings of the problem, or the same representation, but using different subset of features to obtain the partitions. Then, which representation of the data should be used in the QACE step?
2. Besides, the partitions in the cluster ensemble could be obtained by using different similarity (dissimilarity) measures but, which one should be used in the QACE step? We would possibly also need to compute a distance between objects in this step but, what can be done if we have not a distance defined for our set of objects?

These problems are more complicated when the original data is mixed, because it is difficult to apply cluster validity indexes to the partitions since it is difficult to embed the set of objects into a metric space. In this paper, we propose a solution for these questions and it is incorporated in the steps of the WKF algorithm.

3 Generalized WKF

Firstly, we say that the fact that two objects belong to the same cluster in a partition does not contribute with the same information for every partitions in the cluster ensemble. For that reason, we will define the *co-association significance* of two objects x_i and x_j that belong to the same cluster in some partition $P \in \mathbb{P}$. To compute this significance, we will take three factors into account: the number of elements in the cluster to which x_i and x_j belong, the number of clusters in the partition analyzed and the similarity (dissimilarity) of this objects by using the same proximity measure used to conform the partition P. Then, we say that two objects x_i and x_j, grouped in a cluster C of some partition $P \in \mathbb{P}$, which was obtained using the similarity measure $\mathit{\Gamma}_P$, have a high co-association significance if the following conditions are satisfied:

1. $|C|$ is small ($|C|$ is the number of elements in the cluster C)
2. $|P|$ is large ($|P|$ is the number of clusters in P)
3. $\mathit{\Gamma}_P(x_i, x_j)$ is large.

If the partition P was obtained by using a dissimilarity measure d_P, we can easily obtain an equivalent similarity measure $\mathit{\Gamma}_P$ by $\mathit{\Gamma}_P = \frac{1}{d_P+1}$. Then from now on, we assume that the clustering algorithm applied to generate the partition P used the similarity measure $\mathit{\Gamma}_P$. Formally, we can define the co-association significance as:

Definition 1. *Given two objects x_i, x_j and a partition P, we define the* co-association significance *of these objects in the partition P as:*

$$CS^P(x_i, x_j) = \begin{cases} \frac{|P|}{|C|} \cdot \mathit{\Gamma}_P(x_i, x_j), & \mathit{if}\ \exists C \in P, \mathit{such\ that}\ x_i \in C, x_j \in C; \\ 0, & \mathit{otherwise} \end{cases}$$

which represents the significance that two objects x_i and x_j had been grouped together in the same cluster or not, in the partition P.

Taking into account the co-association significance of each pair of objects in X, in all partition in $\mathbb{P}$, it is conformed the *Weighted Co-Association Matrix* denoted by WC, where the (i, j) entry of the matrix has the following value:

$$WC_{i,j} = \sum_{k=1}^{m} CS^{P_k}(x_i, x_j) \tag{2}$$

The WC matrix is more expressive than the traditional co-association matrix ([1]), because in the co-association matrix it is only taken into account if the objects are together or not in the same cluster but, the rest of the information given by the partition is not considered. Let us see an illustrative example:

Example 1. Let $\mathbb{P}_X$ be the set of all possible partitions of the set X. We can define the order relationship *nested in* denoted by $\preceq$, where $P \preceq P'$ if and only if, for all cluster $C' \in P'$ there are clusters $C_{i_1}, C_{i_2}, \ldots, C_{i_k} \in P$ such that $C' = \bigcup_{j=1}^{k} C_{i_j}$. The hierarchical clustering algorithms, like the Single Link and Average Link, produce sequences of nested partitions where, if $P \preceq P'$ means that the criterion used to obtain P is stronger than the used to obtain P'. Then, the fact that a pair of objects belong to the same cluster in P, gives more information about the likeness of these objects than the fact that they had been grouped in the same cluster in the partition P'. Using the traditional co-association matrix (1) this information can not be extracted. However, by using the weighted co-association (2) more significance is given to the partition P since, if $P \preceq P'$ implies that $|P| \geq |P'|$ and $|C| \leq |C'|$. Then $\frac{|P|}{|C|} \cdot \mathit{\Gamma}_P(x_i, x_j) \geq \frac{|P'|}{|C'|} \cdot \mathit{\Gamma}_{P'}(x_i, x_j)$ because in this case $\mathit{\Gamma}_P = \mathit{\Gamma}_{P'}$.

Once we have the matrix WC, it can be considered as a new representation of the objects, where each object $x_i \in X$ is represented by a vector of $\mathbb{R}^n$, $x_i = \{WC_{i,1}, WC_{i,2}, \ldots, WC_{i,n}\}$. The representation by dis(similarities) is extensively studied in [4]. This way, all the information about the possible different representations and proximity measures, used in the generation step, are summarized in the new representation of the objects. Even, when only one representation of the set of objects, and only one similarity measure in the generation step are used, this new representation can give advantages, e.g. when the original data is mixed. In this case, the new representation as a vector of $\mathbb{R}^n$ allows the use of the mathematical tools for vectorial spaces that have not to be available for the original data representation.

In [5], a comparison of different cluster ensemble techniques is achieved. Among other techniques, the co-association matrix (1) is used as a new representation of the objects, obtaining the best results. Then, as an alternative, the new weighted co-association matrix can be used instead of the traditional co-association matrix in the co-association cluster ensemble methods [2]. The results should be better since the WC matrix has more information about the real relationship between the objects in the set of partitions $\mathbb{P}$.

However, the main goal of the construction of the matrix WC is for obtaining a new representation of the objects that allows to use the WKF method without any constraint in the generation step.

3.1 Steps of the Generalized WKF Algorithm

In order to embed our object representation into the WKF methodology, it is necessary to incorporate an intermediate step where the matrix WC is computed and used as a new representation of the objects. We call the algorithm with this modification as GWKF and its principal steps are:

1. Given a set of objects X, generate a set of partitions $\mathbb{P}$ of these objects, by using different clustering algorithms, different initialization of the parameters, even using different representation of the objects, and different similarity (dissimilarity) measures to obtain each partition.

2. With the information in $\mathbb{P}$, compute the *weighted co-association matrix* (2), and use it as a new representation of the set of objects X.
3. Apply the QACE, where any kind of indexes can be used without taking into account the original type of data of the problem or the way that the partitions in the cluster ensemble were generated. The indexes are applied by using the new data representation and can be used any distance function (e.g., the Euclidean distance).
4. Compute an associated weight to each partition by using the indexes defined previously.
5. Apply the consensus function as in [1]. This consensus function automatically selects the appropriate number of clusters in the consensus partition.

4 Experimental Results

In the experiments, we used 7 datasets from the UCI Machine Learning Repository [8]. The characteristics of all datasets are given in Table 1. We denote our method (given by the steps in the previous section) by GWKF. Also, in order to apply the QACE step, we use three simple indexes [6]: *Variance*, *Connectivity* and *Dunn Index*. Each one of them measures the fulfillment of a specific property. The *Variance* is a way to measure the compactness of the clusters in the partition. The *Connectivity* evaluates the degree of connectedness of clusters in the partition, by measuring how many neighbors of one object belong to the same cluster that the object. The *Dunn Index* is a measure of the ratio between the smallest inter-cluster distance and the largest intra-cluster distance.

The three indexes use a distance function defined over the set of objects X. Then, if it is used the WKF algorithm, these indexes can not be applied to a dataset for which there is not defined an appropriate distance function. However, by using the GWKF, we can apply this indexes to any dataset without taking into account the type of data because, after the generation step the objects are represented as vectors of $\mathbb{R}^n$ by using the Weighted Co-Association Matrix (2). After that, any distance function can be applied, we use the Euclidean distance.

Table 1. Overview of datasets

Name	Type	#Inst.	#Attrib.	#Classes	Inst-per-classes
Iris	Numerical	150	4	3	50-50-50
Digits	Numerical	100	64	10	10-11-11-11-12-5-8-12-9-11
Breast-Cancer	Numerical	683	9	2	444-239
Zoo	Mixed	101	18	7	41-20-5-13-4-8-10
Auto	Mixed	205	26	7	0-3-22-67-54-32-27
Soybeans	Categorical	47	21	4	10-10-10-17
Votes	Categorical	435	16	2	267-168

4.1 Analysis of the Experimental Results

Firstly, we show the behavior of our method (GWKF) in numerical datasets and we compare the results with several clustering ensemble methods (see Table 2). EA-SL and EA-CL are the co-association methods [2] using the Single-Link and Complete-Link algorithm, CSPA, HPGA and MCLA are the three hypergraph methods proposed in [3]. In this experiment, it is generated the cluster ensemble always using the same representation of the objects, and by applying the k-Means algorithm 20 times with different parameters initialization. This experiment shows the good performance of the GWKF method in comparison with the other cluster ensemble methods and how the final results of the GWKF method are very close to the results of the WKF method.

Table 2. Clustering error rate of different clustering ensemble methods

Dataset	**Ens(Ave)**	**EA-SL**	**EA-AL**	**CSPA**	**HPGA**	**MCLA**	**WKF**	**GWKF**
Iris	18.1	11.1	11.1	13.3	37.3	11.2	**10.6**	10.8
Breast-Cancer	3.9	4.0	4.0	17.3	49.9	3.8	**3.7**	**3.7**
Digits	27.4	56.6	23.2	**18.1**	40.7	18.5	22.1	20.6

The WKF method can not be applied if the data is mixed or in the generation step were used different representations of the objects. In these cases, the GWKF method gains more importance, because it is able to deal with any type of data and any kind of generation mechanism keeping the good performance of the WKF method.

On the other hand, in Table 3, we compare the GWKF with simple clustering algorithms (in this case the k-Means) in two different mixed datasets with different ensemble sizes (H). In this experiment, the partitions are obtained by using random subset of features and applying the k-Means algorithm with a fixed number of clusters (k). The results show that our algorithm obtains lower errors rate than the average error rate of the k-Means algorithm.

Table 3. Cluster ensemble average error rate and GWKF error rate in mixed datasets

Dataset	**H**	**k**	**Ensemble(Ave.)**	**GWKF**
Zoo	10	7	26.8	**20.7**
Zoo	20	7	24.3	**19.1**
Auto	10	7	62.0	**57.3**
Auto	20	7	61.3	**54.5**

Finally, we compare our method with 4 algorithms proposed in [7] (CSPA-METIS, CSPA-SPEC, CSBA-METIS and CSBA-SPEC), all of them designed to work with categorical data. In this experiment, we use 2 categorical datasets, and the partitions were generated by using different sets of random features (see Table 4). As in the previous experiment, the original WKF method is not able to

Table 4. Clustering error rate using categorical data

Dataset	k	Ens(Ave)	CSPA-ME	CSPA-SP	CBPA-ME	CBPA-SP	GWKF
Votes	2	13.7	14.0	13.5	14.0	14.2	**11.4**
Soybeans	4	24.4	10.6	12.3	12.8	15.3	**6.3**

work with this generation mechanism, this dataset and the indexes defined for the experiments. However, the GWKF method extends the good performance of the WFF method to this kind of situations.

5 Conclusions

In this paper, the problem about what representation of the objects and what similarity measure should be used in the consensus step, in the cases that many representations and similarity measures are used in the generation step is formulated and solved. A new object representation is proposed, which summarizes the whole information in the set of partitions. This is possible thanks to the definition of a new way of measuring the co-association between objects, which is more expressive about the real similarity between objects in the cluster ensemble than the classical co-association. The new representation is embedded in the WKF method enlarging its scope and obtaining the Generalized WKF method. The experiments with numerical, categorical and mixed datasets respectively and by using different generation mechanisms, show the good performance of the proposed method.

References

[1] Vega-Pons, S., Correa-Morris, J., Ruiz-Shulcloper, J.: Weighted cluster ensemble using a kernel consensus function. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 195–202. Springer, Heidelberg (2008)

[2] Fred, A.L.N., Jain, A.K.: Combining multiple clustering using evidence accumulation. IEEE Trans. on Pat. Analysis and Machine Intelligence 27, 835–850 (2005)

[3] Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617 (2002)

[4] Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations And Applications. In: Machine Perception and Artificial Intelligence. World Scientific Publishing Co., Singapore (2005)

[5] Kuncheva, L., Hadjitodorov, S., Todorova, L.: Experimental comparison of cluster ensemble methods. In: Int. Conference on Information Fusion, pp. 1–7 (2006)

[6] Handl, J., Knowles, J., Kell, D.: Computational cluster validation in post- genomic data analysis. Bioinformatics 21, 3201–3212 (2005)

[7] Al-Razgan, M., Domeniconi, C.: Random subspace ensembles for clustering categorical data. Studies in Computational Intelligence (SCI) 126, 31–48 (2008)

[8] UCI machine learning repository, http://archive.ics.uci.edu/ml/datasets.html

Using Maximum Similarity Graphs to Edit Nearest Neighbor Classifiers

Milton García-Borroto[1,3], Yenny Villuendas-Rey[2], Jesús Ariel Carrasco-Ochoa[3], and José Fco. Martínez-Trinidad[3]

[1] Bioplantas Center, UNICA, Carretera a Morón km 9 ½, C. de Ávila, Cuba
mil@bioplantas.cu
http://www.bioplantas.cu
[2] Ciego de Ávila University UNICA, Carretera a Morón km 9 ½, C. de Ávila, Cuba
yennyv@bioplantas.cu
http://www.unica.cu
[3] National Institute of Astrophysics, Optics and Electronics, Puebla, México
{ariel,fmartine}@inaoep.mx
http://www.inaoep.mx

Abstract. The Nearest Neighbor classifier is a simple but powerful non-parametric technique for supervised classification. However, it is very sensitive to noise and outliers, which could decrease the classifier accuracy. To overcome this problem, we propose two new editing methods based on maximum similarity graphs. Numerical experiments in several databases show the high quality performance of our methods according to classifier accuracy.

Keywords: nearest neighbor, error-based editing, prototype selection.

1 Introduction

One of the most popular non-parametric classifiers is the Nearest Neighbor (NN). This classifier combines the simplicity with a classification error bounded by twice the Bayes error [1]. For classifying a new object, the NN classifier compares it against all the objects in the training set, and assigns the new object to the class of its nearest object.

An important drawback of the NN classifier is its sensitivity to noisy and mislabeled objects [2]. Since NN introduction in 1967, there is a constant research interest in the Pattern Recognition community to overcome this drawback [3-5]. Editing algorithms aim at improving classifier accuracy by deleting noisy or mislabeled objects.

In this paper, we address the problem of improving NN accuracy by smoothing classification boundaries. We use maximum similarity graphs to determine border objects, and delete those noisy or mislabeled objects that most likely could affect the classifier accuracy.

This paper is organized as follows: section two describes some previous works about NN error-based editing and their drawbacks, section three introduces our proposals, section four shows the results of the experiments and section five offers some conclusions.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 489–496, 2009.

2 Previous Works on Error-Based Editing

Several authors divide the algorithms to improve a training set for NN classifiers in two main categories: condensing algorithms and error-based editing algorithms (or just editing algorithms) [6]. Condensing algorithms aim at reducing the NN computational cost by obtaining a small subset of the training set, maintaining the accuracy as high as possible, while editing algorithms aim at improving classifier accuracy by deleting noisy or mislabeled objects. In this work, we focus on editing algorithms.

The first editing algorithm was the Edited Nearest Neighbor (ENN), proposed by Wilson in 1972 [3]. The ENN algorithm deletes all objects misclassified by a k-NN classifier, where k is a user-defined parameter, usually $k = 3$. The results of ENN strongly depend on the value of k, and there is not a simple procedure to find this value *a priori.* Another weak point is the use of a unique value of k for the entire database; without taking into account the object densities in different regions. Finally, a major drawback of ENN comes from the behavior you can see in Fig. 1. Note that the border object A will be removed using even a low value of $k = 3$, no matter it is very similar to other objects of its own class.

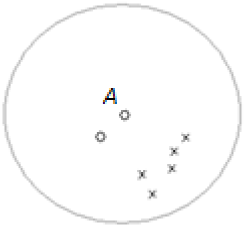


Fig. 1. ENN can erroneously remove objects in the class boundaries

In 1976, Tomek introduced the All-KNN algorithm[4]. All-KNN deletes an object if a k-NN classifier misclassifies it, with k in the range $1 \leq k \leq kMax$ where $kMax$ is a user-defined parameter, usually $kMax = 7$ or $kMax = 9$. The use of several values for k in All-KNN makes the algorithm to do more deletions than ENN. Nevertheless, in many cases, like that showed in Fig. 1, it produces an undesired behavior. All-KNN keeps the same drawbacks than ENN.

Another classical editing method is MULTIEDIT, proposed by Devijver and Kittler in 1980 [5]. First, MULTIEDIT randomly divides the training set in ns partitions. On each partition, it applies the ENN method using a 1-NN classifier trained with the next partition. After each iteration, MULTIEDIT joins the remaining objects in each partition and it repeats the process until no change is achieved in ni successive iterations. Both ns and ni are user-defined parameters. Usually $ns = 3$ and $ni = 2$. This method can successfully purge noisy objects and outliers, but if two classes are very close, it can completely remove one or both of them (Fig. 2). Also, the strong random characteristic of MULTIEDIT could make the result of two consecutive executions over the same training set to be completely different, as you can also see in Fig. 2.

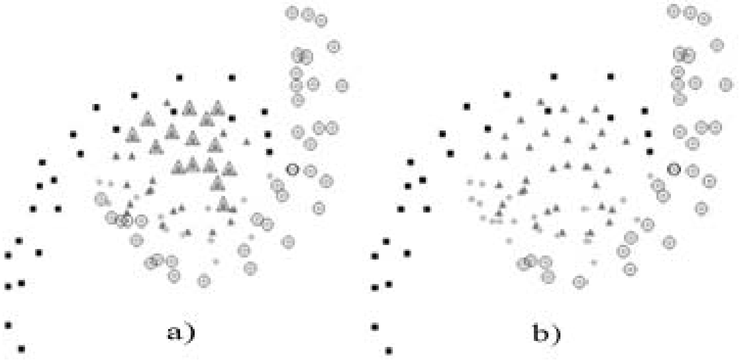


Fig. 2. Example of two MULTIEDIT executions: (a) one class deletion and (b) two class deletion. ■ ● ▲ are the objects in the training matrix; □ ○ △ highlight the selected objects.

Hattori and Takahashi in 2000 [7] proposed a new editing method, referred by us as NENN. The method computes the k nearest neighbors for each object. If an object has at least one of its neighbors in another class, NENN deletes this object from the training set. This condition is stronger than the one used in ENN, and could produce more dramatic object deletion. Unfortunately, if the boundaries between classes are close together, it can delete many important boundary objects, like the whole " ○ " class in Fig. 1.

In 2002, Toussaint used the Relative Neighborhood proximity graphs to edit nearest neighbors [8]. The RNG-E algorithm computes the Relative Neighborhood graph of the training set, and deletes all objects misclassified by its neighbors in the graph. The Relative Neighborhood graph is a proximity graph with the set of edges defined as $ges = \{(p, q): \Lambda_{\mathrm{p,q}} \cap T = \emptyset\}$, where p and q are vertexes, T is the training set, and $\Lambda_{\mathrm{p,q}}$ is the intersection between the hyper-spheres centered in p and q respectively, with radius $||p - q||$.

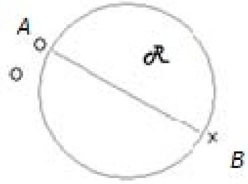


Fig. 3. A common problem of RNG-E algorithm

A common problem with RNG-E is that the proximity graph can connect faraway objects, because RNG-E depends on the configuration of other objects in the graph. For example, in Fig. 3 you can see that objects A and B are neighbors because no other object exists in the region $\mathcal{R}$ even though they are distant.

Caballero *et al.*, introduced the editing methods EditRS1 and EditRS2 in 2007 [9]. They used elements of the Rough Set theory to obtain lower and upper approximations of the training set for computing the limit regions of each class. The EditRS1 algorithm computes the lower approximation of each class and deletes the objects not included in the lower approximation. The EditRS2 also computes the lower approximation of the classes, and the limit region or border of each class. For each limit region, EditRS2 applies the Generalized Editing method [10]. Finally, the method deletes such objects not included in the lower approximations or in the edited limit region. Although both methods are supported by a well-founded theory, in most of the tested databases they were unable to remove any object.

The analysis of previous editing methods reveals that improving the k-NN accuracy by editing the training set is still an open problem. Previous methods cannot accurately deal with some cases, which is the basic motivation of the methods introduced in this paper.

3 Editing Based on Maximum Similarity Graphs

In this paper, we introduce MSEditA and MSEditB, two new editing methods based on maximum similarity graphs (MSG). A maximum similarity graph [11] is a directed graph, where each object is connected to its most similar neighbor. Formally, let $G = (T, \theta)$ be a maximum similarity graph for a training set T, with arcs θ. Two objects $x, y \in T$ form an arc $(x, y) \in \theta$ if

$$\max_{o \in T}\{sim(x, o)\} = sim(x, y)$$

where $sim(x, y)$ is a similarity function, usually defined in terms of the normalized Euclidean distance as $1 - d(x, y)$, but in general it can be any similarity or dissimilarity. This way we have an arc between an object and its most similar neighbor. If we have ties, we insert arcs for all the most similar neighbors.

A MSG is very useful for object selection because it can catch the similarity relations between any object and those that are on its neighborhood. It is also immune to configurations like that appearing in Fig. 1, which makes most of other editing methods to fail. Although in theory MSG can connect faraway objects, it is not a common behavior, because both objects have to be in complete isolation. In the example in Fig. 3, for example, the MSG does not connect the objects A and B.

3.1 Proposed Methods

Both MSEditA and MSEditB methods first compute the maximum similarity graph of the training set T, and then decide which objects to delete. In MSEditA, an object in the graph having a most similar neighbor from different class indicates a level of uncertainty of the correct class for the object. The idea followed in MSEditA consists in deleting an object if it has any most similar neighbor (successors) with different class. This method is different from ENN (using $k = 1$), because if the most similar object is not unique, ENN randomly selects one of them. In this case, MSEditA discard the object if any of the neighbors belongs to a different class.

On the other hand, MSEditB removes an object if the majority of its neighbors belong to a different class. We count as neighbors both successors and predecessors of an object in the graph, which are respectively its most similar objects and the objects for which the evaluated object is the most similar.

Pseudocode of MSEditA

```
1. $Edited \leftarrow T$
2. Compute maximum similarity graph of $T$.
3. For each object in $Edited$
4. If the object has at least one successor of different class in the maximum similarity
   class, delete the object.
5. Return $Edited$.
```

Pseudocode of MSEditB

```
1. $Edited \leftarrow T$
2. Compute maximum similarity graph of $T$
3. For each object $o$ in $Edited$
4. Let be $S_o$ and $A_o$ the successors and antecessors of $o$ in the graph.
5. $N \leftarrow S_o \cup A_o$
6. If the majority of the objects in $N$  are not of the same class of $o$, delete $o$.
7. Return $Edited$
```

We can see the differences between MSEditA and MSEditB in Fig. 4.

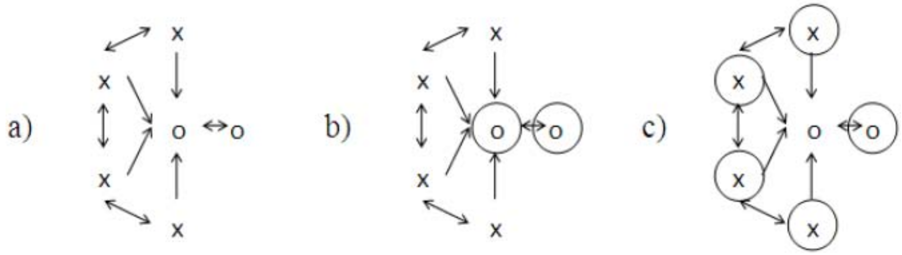


Fig. 4. Diferences (circled) between MSEditA (b) and MSEditB (c) on example MSG (a)

There are six objects from two classes: circle and cross. Each arrow represents an arc, so a single arrow from A to B means the successor B is the most similar object to the ancestor A, and a double arrow means they are simultaneously the most similar object of each other.

MSEditA deletes all crosses, because they have one of their successors in a different class, while MSEditB only deletes the central object, because most of its neighbors (successors and ancestors) are in a different class. It is important to notice that the results of these methods are different, and it is easy to prove that, in general, the result of a method is not contained in the other.

4 Experimental Results

In order to compare the performance of the proposed methods, we carried out some numerical experiments in a wide-range of databases from the UCI repository of Machine Learning [12]. The description of the tested databases appears in Table 1. We perform 10-fold cross validation among all databases, averaging the results. Due to NN classifier is dependant on the function used for comparing objects, in our experiments we use two functions: HEOM and HVDM [13].

We select several classic, state of the art and recent methods, which are frequently used for comparisons in most papers about the topic. For determining the best method, we made a two-tailed T-Test [14] (with significance of 0.05) of every method with respect to the lowest error result. If no significant difference exists, the result is also considered as lowest error. We compute how many databases each method attains the lowest error. The object retention ratio is calculated as the ratio between the amount of objects in the edited sample and the amount of objects in the training sample. Table 2 summarizes the results with both HEOM and HVDM functions.

Table 1. Databases description

Database	non-num/ num feats	Objects	Database	non-num/ num feats	Objects
anneal	29/9	798	hepatitis	13/6	155
autos	10/16	205	iris	0/4	150
balance-scale	0/4	625	labor	6/8	57
breast-cancer	9/0	286	lymph	15/3	148
breast-w	0/9	299	post-pat-data	7/1	90
cmc	7/2	1473	sonar	0/60	208
colic	15/7	368	tae	2/3	151
credit-a	9/6	690	trains	29/4	10
cylinder-bands	20/20	512	vehicle	0/18	846
dermatology	1/33	366	vote	16/0	435
glass	0/10	214	vowel	3/9	990
heart-c	7/6	303	zoo	16/1	101
heart-h	7/6	294			

In *tae* and *vowel* databases, using both HEOM and HVDM functions, all methods produce a significant degradation of the accuracy, which implies that all objects are very important to maintain classification accuracy. The same situation occurred in *cylinder-bands* database for the HEOM function. Both MSEditA and MSEditB have the best result, having the lowest error, in 22 of 25 databases. In general, usually one of the methods (not always the same) gets better results than the other gets, but we need further research to allow selecting *a priori* the best method for a database. It is important to highlight that some methods tie in most databases, so there is no categorical winner.

Table 2. Number of databses (from 25 databases) each method had lower error (err) and object retention ratio (ret)

	ENN	**All-KNN**	**MULTIEDIT**	**NENN**	**RNG-E**	**EditRS1**	**EditRS2**	**MSEditA**	**MSEditB**	**Original**
err HEOM	21	19	15	18	21	21	21	**22**	21	**22**
err HVDM	22	20	14	17	22	22	22	22	**23**	22
Average err	21.5	19.5	14.5	17.5	21.5	21.5	21.5	**22**	**22**	**22**
ret HEOM	0.88	0.74	**0.55**	0.57	0.76	1.00	1.00	0.74	0.75	
ret HVDM	0.88	0.74	**0.57**	0.60	0.77	1.00	1.00	0.77	0.75	

According to the object retention ratio (see Table 2), we show the averaged result for both HEOM and HVDM functions. In *cmc* database, for both functions, the NENN method deleted all objects, so we did not average this result. The best method according to retention rate was MULTIEDIT, but it was the worst according to classification accuracy. Both EditRS1 and EditRS2 only reduce objects in *breast-cancer* and *tae* databases, in all other databases they did not delete any object of the training set.

5 Conclusions

In this paper, we introduce two new editing methods based on maximum similarity graphs. Both methods have the best results according to classifier accuracy in the tested databases, by deleting noisy and mislabeled objects. They attain the higher accuracy in 22 of 25 databases. MSEditA and MSEditB have object retention rate around 75%, comparable with other editing methods such as RNE-G and All-KNN. Although NN classifier depends on the function used for comparing objects, we did not found significant difference in the performance of the methods with both HEOM and HVDM functions.

As future work, we are going to study which are the characteristics of the database that makes one of our methods to overcome the other. This way, we will be able to create a combined method with even better behavior.

Acknowledgments

The first author wants to thank to the National Institute of Astrophysics, Optics and Electronics for its support on this research. This work is partly supported by National Council of Science and Technology of México under grant 252752.

References

1. Cover, T., Hart, P.E.: Nearest Neighbor pattern classification. IEEE Trans. on Information Theory 13, 21–27 (1967)
2. Dasarathy, B.D.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
3. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics 2, 408–421 (1972)
4. Tomek, I.: An experiment with the Edited Nearest-Neighbor Rule. IEEE Transactions on Systems, Man and Cybernetics SMC-6, 448–452 (1976)
5. Devijver, P.A., Kittler, J.: On the edited neares neighbor rule. In: Press, I.C.S. (ed.) 5th International Conference on Pattern Recognition, Los Alamitos, California, pp. 72–80 (1980)
6. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. Wiley-Interscience, Hoboken (2004)
7. Hattori, K., Takahashi, M.: A new edited k-nearest neighbor rule in the pattern classification problem. Pattern Recognition 33, 521–528 (2000)
8. Toussaint, G.: Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress. In: 34 Symposium on Computing and Statistics INTERFACE-2002, Montreal, Canada, pp. 1–20 (2002)
9. Caballero, Y., Bello, R., Salgado, Y., García, M.M.: A method to edit training set based on rough sets. International Journal of Computational Intelligence Research 3, 219–229 (2007)
10. Koplowitz, J.: On the relation of performance to editing in nearest neighbor rules. Pattern Recognit. 13, 251–255 (1981)
11. Pons-Porrata, A., Berlanga-Llavori, R., Ruiz-Shulcloper, J.: Topic discovery based on text mining techniques. Information Processing & Management 43, 752–768 (2007)
12. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases. University of California at Irvine, Department of Information and Computer Science, Irvine (1998)
13. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
14. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, vol. 10, pp. 1895–1923. MIT Press, Cambridge (1998)

A New Incremental Algorithm for Overlapped Clustering

Airel Pérez Suárez[1,2], José Fco. Martínez Trinidad[1],
Jesús A. Carrasco Ochoa[1], and José E. Medina Pagola[2]

[1] National Institute for Astrophysics, Optics and Electronics (INAOE), Mexico
[2] Advanced Technologies Application Center (CENATAV), Cuba
{airel,fmartine,ariel}@ccc.inaoep.mx, jmedina@cenatav.co.cu

Abstract. In this paper, a new algorithm for incremental overlapped clustering, called Incremental Clustering by Strength Decision (ICSD), is introduced. ICSD obtains a set of dense and overlapped clusters using a new graph cover heuristic while reduces the amount of computation by maintaining incrementally the cluster structure. The experimental results show that our proposal outperforms other graph-based clustering algorithms considering quality measures and also show that ICSD achieves a better time performance than other incremental graph-based algorithms.

1 Introduction

Clustering is one of the most useful and common techniques in pattern recognition and data mining. Nevertheless, most clustering algorithms are non-incremental [1,2,3,4], which means that if new data is added to a training set, the algorithm must be applied again over the whole training set, without taking advantage of the previous clusters.

In environments like the World Wide Web, news streams and others, the data must be organized in overlapped clusters and these clusters must be frequently updated due to new data are continually added. Therefore, the problem of incremental overlapped clustering is addressed in this work.

The *graph-based* algorithms have received attention by the researchers in the last years because this kind of algorithms have features that increase their suitability for many applications where overlapped clustering is needed and they also have shown in this context a better performance that commonly used algorithms [1,2,3].

In this paper, a new incremental overlapped clustering algorithm named Incremental Clustering by Strength Decision (ICSD) is introduced. The novelty of the proposed algorithm is its ability to obtain a set of dense and overlapped clusters using a new graph cover heuristic while it reduces the amount of computation by maintaining clusters structured incrementally. The experimental evaluations showed that ICSD algorithm outperforms other graph-based algorithms [4,5,6].

The remainder of this paper is organized as follows. Section 2 presents related work. The ICSD algorithm is presented in section 3 and finally, conclusions and future work are given in section 5.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 497–504, 2009.

2 Related Work

There are different graph-based algorithms developed during the last years that deal with incremental clustering [5,6,7,8] or with overlapped clusters [1,2,3,4,5], but only a few of them faced both problems at the same time [5,6].

From the group of algorithms that build overlapped clusters [1,2,3,4,5], Cstar [4] has the best performance and outperforms two commonly used clustering algorithms: Single-link and UPGMA algorithms [9]. However, Cstar algorithm only groups static data and it has a computational complexity of O(n^3).

Among all the incremental graph-based algorithms [5,6,7,8] only Star [5] and Strong Compact [6] algorithms faced the problem of overlapped clusters in an incremental context, while GLC[8] and Compact[7] build disjoint clusters; however, all these incremental algorithms have different drawbacks. The set of clusters built by GLC algorithm are connected components so they could have low cohesion. The Star, Compact and Strong Compact algorithms, on the other hand, build a lot of clusters, each one with a few prototypes. Additionally, Star algorithm builds clusters that depends on data order.

Star and Strong Compact algorithms, no matter how many prototypes are added to the dataset, update the set of clusters by adding those prototypes one by one and and they update the clusters after each addition. This is a constraint when a lot of new data must be added at the same time, since it diminishes the performance of these algorithms.

The algorithm proposed in this work introduces a new graph cover heuristic that produces overlapped clusters with high density. Our algorithm keeps the strengths of Cstar algorithm but with a lower computational complexity. The proposed algorithm also allows an efficient cluster update; in this way, the processing time is reduced.

3 Clustering by Strength Decision

The ICSD algorithm, introduced in this section, obtains a set of overlapped clusters through a cover of the *thresholded similarity graph* G_β by applying an heuristic based on the *strength* of vertices in G_β and using *star-shaped subgraphs*[5].

Let $D = \{p_1, p_2, \ldots, p_n\}$ be a collection of prototypes, β a user-defined parameter and $S(p_i, p_j)$ a symmetric similarity function between prototypes p_i and p_j, a *thresholded similarity graph* is an undirected graph $G_\beta = \langle V, E_\beta \rangle$ where $V = D$ and $(p_i, p_j) \in E_\beta$ if and only if $S(p_i, p_j) \geq \beta$.

An *star-shaped sub-graph* is a sub-graph of $m + 1$ vertices with an special vertex c named *center* and m vertices called *satellites*. This sub-graph satisfies that there is an edge between the center and each satellite. When a star-shaped sub-graph only contains its center it is called *degenerated*. In this context, each star-shaped subgraph is interpreted as a cluster.

A cover of G_β, by this kind of sub-graphs, is determined by the set of centers C such that every vertex of G_β belongs to C or it is adjacent to at least one vertex in C. Usually, to obtain a cover of G_β, a greedy approach is applied [1,2,3,4,5].

The *strength* of a vertex v is calculated in two steps: in the first step, a vertex v receives one vote from each vertex u such that:

$$u \in v.Adj \wedge v.degree \geq u.degree$$

where $v.Adj$ is the set of adjacent vertices of v and $v.degree$ is the number of vertices contained in $v.Adj$ and $u.degree$ is defined in the same way that $v.degree$. The number of votes received by v in this first step will be denoted as $strength_{pre}$.

In a second step, $v.strength$ is calculated as follows:

$$v.strength = |\{u \in v.Adj \mid v.strength_{pre} \geq u.strength_{pre}\}|$$

The heuristic proposed in this work for building overlapped clusters considerers only the set Q of vertices with a non null value of strength; in this way, the set of vertices to be possibly included in C is reduced.

The set Q is sorted in descending order according to the strength and it is processed in that order for covering G_β as quick as possible. Each vertex $v \in Q$ is processed considering the following conditions:

1. if v is not covered in G_β then, it is added to the set of centers,
2. if v is covered in G_β then add v to C if and only if it has at least one not covered adjacent vertex. This condition raises from the fact that as overlap is allowed there would be vertices in Q which have all their satellites covered by previous selected vertices; thus selecting those vertices as center will not cover new vertices in G_β, therefore, they must not be included in C.

Once the set C has been built, a process, similar to that used by Cstar [4] to remove *redundant centers*, but using the degree of vertices instead of their voting-degree, is executed.

We decided to use degree of verties because of two main reasons: (a) since a cluster is defined by a center and its satellites then, the more degree a center has the more density[1] its cluster has, and (b) the vertex v, of a set of vertices M, with the highest voting-degree not always forms the densest cluster, because there could be another vertex $g \in M$, having a higher degree than v, which received a lower voting degree due to most of its adjacent vertices gave their vote to another adjacent vertex.

Finally, remaining vertices in C are marked as *center* and all other vertices in G_β are marked as *satellites*. As result, we will have a set of overlapped clusters, where the set of clusters is built from each vertex marked as center and its satellites.

Assuming that we have a set of overlapped clusters built using the heuristic presented above. Now we will analyze the clusters that are affected when new vertices or prototypes are added.

Let $G' = \langle V', E' \rangle$ be a connected component of G_β and $v \in V$, we will say that v *generates* the component G' if and only if $v \in V'$. A cluster should be updated

[1] In this paper density is defined as the average number of elements per cluster.

if its center, or at least one of its satellites, changes its strength after adding a new vertex. A vertex could changes its strength value only if it belongs to the component generated by some added vertex; therefore, only those components need to be re-clustered.

To update the clusters contained in a connected component it is important to know, first of all, how the insertions affect the current clustering built through the above described heuristic.

After some vertices are added to G_β the following could happen:

a) there are uncovered vertices; therefore, new vertices must be added to set C.
b) there is at least one non center vertex v which has an strength value greater than one of its adjacent centers or one center $c \in C$ that covers some vertices in $v.Adj$. All vertices like v should be considered to be included in C because they could improve the clustering's density.

For updating the clusters of a connected component $G' =< V', E' >$ the set of vertices V' is divided in the sets V'_s and V'_c which contain the set of all satellites having strength greater than zero and the set of all centers, respectively. Each one of these sets is processed independently to determine which vertices must be added to the candidate list Q.

Each satellite $s \in V'_s$ is processed considering the following conditions:

a) if s is uncovered or has at least one uncovered adjacent vertex then s is inserted into Q.
b) if s has at least one adjacent vertex v such that $s.strength > w.strength$, where w is the center having the higher value of strength among all adjacent centers of v then, s is inserted into Q and v is marked as *activated*; vertices marked as *activated* are useful when set V_c is processed.

Each center $c \in V'_c$ is processed considering the following conditions:

a) for each $v \in c.Adj$ such that $v.strength > c.strength$, insert v into Q and mark c as *weak*.
b) if c is marked as *weak* or it has at least one adjacent vertex marked as *activated* then, c is removed from V_c, it is marked as satellite and if $c.strength > 0$, c is inserted into Q.

Once the set Q has been built for the connected component G', it is processed using the heuristic proposed above for building overlapped clusters.

The pseudocode of ICSD is showed in Algorithm 1.

ICSD algorithm has a computational complexity of $O(n^2)$ (the proof was omitted due to space restrictions). ICSD unlike Star, Compact and Strong Compact algorithms, adds all new incoming vertices to G_β before updating the set of clusters, and it also applies an heuristic which only processes the clusters that have been actually affected due to insertions. These characteristics of ICSD make the algorithm to save time making it able to efficiently manage multiple insertions of prototypes.

Algorithm 1. ICSD algorithm

```
Input: $G_\beta$ - a thresholded similarity graph
       $L$ - set of incoming prototypes
       $\beta$ - a similarity threshold
Output: $G_\beta$ - updated thresholded similarity graph
        $SC$ - set of overlapped clusters
"Add each vertex in $L$ to $G_\beta$ and update $G_\beta$";
foreach new added vertex $v$ do
    if $v$ is marked as not-processed then
        "Build the connected component $G' =< V', E' >$ associated with $v$";
        if $G'$ is an isolated vertex then "Mark $v$ as center";
        else
            "calculate strength property for each vertex in $G'$";
            "Build $V'_s, V'_c$, and $Q$ sets";
            while $Q \neq \emptyset$ do
                $u := \arg max_x \{x.strength \mid x \in Q\}$;
                if $u$ satisfies conditions 1) or 2) then "Add $u$ to $V'_c$";
                $Q := Q \setminus \{u\}$;
            end
            "Sort $V'_c$ in ascending order by degree";
            "Remove from $V'_c$ all redundant centers";
            "Mark vertices in $V'_c$ as center and vertices in $V' \setminus V'_c$ as satellite";
        end
        "Mark vertices in $V'$ as processed";
    end
end
"Mark vertices in $V$ as not-processed";
"Build set $SC$";
```

4 Experimental Evaluation

In this section, an experimental evaluation of the proposed algorithm is presented. Since ICSD algorithm deals with overlapping clustering, the experiments were done over document collections where, as it was mentioned, some documents could belong to more than one cluster.

The document collections used in our experiments were extracted from three benchmark text collection: TREC-5, *Reuters*-21578 and TDT2. From these benchmarks, six document collections were formed: (1) AFP, built from TREC--5; (2) Reu-Te, built using the documents in *Reuters*-21578 tagged as "Test"; (3) Reu-Tr, built using the documents in *Reuters*-21578 tagged as "Train"; (4) Reu-To is the union of Reu-Te and Reu-Tr; (5) TDT2-v1 and (6) TDT2-v2 are two sub-collections of TDT2. The characteristics of all these collections are summarized in Table 1.

In the experiments, documents are represented using the Vector Space model where index terms represent the lemmas of words appearing in the collection; Stop words were removed. The terms of each document were statistically

Table 1. Characteristics of document collections

Collection	Documents	Topics	Overlapping	Terms
AFP	695	25	1.02	11785
Reu-Te	3587	100	1.30	15113
Reu-Tr	7780	115	1.24	21901
Reu-To	11367	120	1.26	27083
TDT2-v1	8603	176	1.17	51764
TDT2-v2	10258	174	1.19	53706

weighted using the logarithm of term's frequency. The cosine measure was used as similarity function.

For the experiments presented in this section, two measures commonly used to evaluate overlapped clustering were selected: Fmeasure (Fme) [10] and Jaccard-index (Jindex) [11]. Both measures evaluate quality based on how much the clustering resembles a set of classes manually labeled by experts; the higher the value of each measure is the better the clustering is.

The experiments were focused on comparing, through Fmeasure and Jaccard-index, the set of clusters obtained by Strong Compact (SComp), Star and Cstar algorithms against the clusters built by ICSD algorithm; Although Cstar is a non incremental algorithm, we decided to include it in this first experiment because it obtained the best quality results for overlapped clustering in different works [3,4]. Table 2 shows the best value of Fmeasure and Jaccard-index obtained by each algorithm for β values in [0.15,0.75].

Table 2. Results for each document collection

		AFP				*Reu-Te*				*Reu-Tr*			
Measures		SComp	Star	Cstar	ICSD	SComp	Star	Cstar	ICSD	SComp	Star	Cstar	ICSD
Fme	value	0.10	0.73	0.76	0.76	0.01	0.57	0.63	0.64	<0.01	0.56	0.56	0.57
	β	0.15	0.25	0.25	0.25	0.15	0.25	0.25	0.25	0.15	0.20	0.25	0.25
Jindex	value	0.05	0.57	0.61	0.61	0.01	0.40	0.46	0.47	<0.01	0.39	0.39	0.40
	β	0.15	0.25	0.25	0.25	0.15	0.25	0.25	0.25	0.15	0.20	0.25	0.25

		Reu-To				*TDT2-v1*				*TDT2-v2*			
Measures		SComp	Star	Cstar	ICSD	SComp	Star	Cstar	ICSD	SComp	Star	Cstar	ICSD
Fme	value	0.01	0.57	0.58	0.59	0.01	0.39	0.44	0.45	0.01	0.44	0.52	0.52
	β	0.15	0.20	0.25	0.25	0.15	0.30	0.30	0.30	0.15	0.30	0.30	0.30
Jindex	value	<0.01	0.40	0.41	0.41	0.01	0.24	0.28	0.29	<0.01	0.28	0.35	0.35
	β	0.15	0.20	0.25	0.25	0.15	0.30	0.30	0.30	0.15	0.30	0.30	0.30

As it can be noticed from Table 2, ICSD algorithm outperforms all the other algorithms in almost all cases.

Fig. 1 shows the results of a second experiment done in order to show the time spent by ICSD, Star and SComp, to cluster the largest tested collection (*Reu-To*) in an incremental way. In Fig. 1(A), curves ICSD, Star and SComp represent the time spent by each algorithm to update the clusters each time 1000

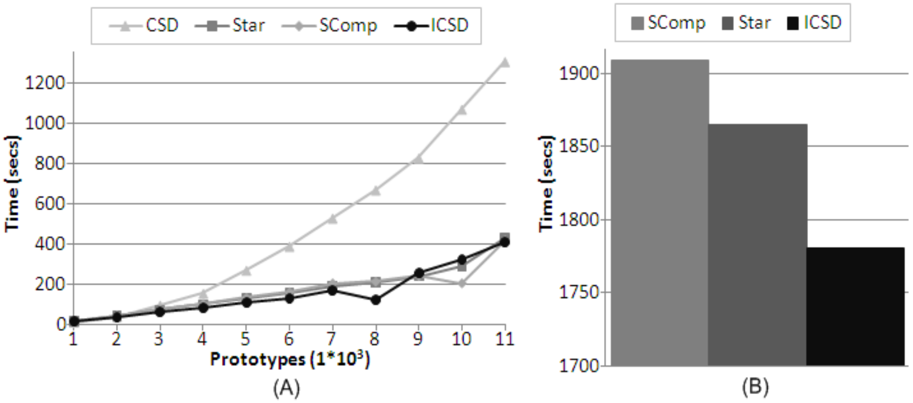


Fig. 1. (A) Behavior of incremental algorithm , B) total time for clustering the whole dataset in an incremental way

Table 3. Comparison considering number and density of clusters

	AFP		*Reu-Te*		*Reu-Tr*		*Reu-To*		*TDT2-v1*		*TDT2-v2*	
Alg.	Grp	Dty	Grp	Dty	Grp	Dty	Grp	Dty	Grp	Dty	Grp	Dty
SComp	413	3.4	1981	3.5	4366	3.6	6437	3.5	4834	3.5	5587	3.5
Star	54	29.1	157	110.8	203	224.2	255	294.4	241	278.8	254	331.5
Cstar	41	68.0	101	523.9	132	1420.5	177	1980.7	168	3349.5	172	3864.6
ICSD	41	69.9	104	546.2	136	1452.0	178	2030.6	172	3401.4	175	3938.5

documents are added[2] and curve CSD represents the time spent to cluster all the prototypes from scratch. Fig. 1(B) shows the total time spent by ICSD, Star and SComp to cluster incrementally the entire dataset; CSD was not included in this figure to avoid scale problems.

As it can be observed, ICSD overcomes Star and SComp and also it scales well in comparison with a non incremental clustering. A similar behavior was observed in the other datasets.

Finally, as SComp is the algorithm which produces the large number of clusters, we selected the β value (for building G_β) for which SComp obtained the smallest number of clusters and we compared the number and density of those clusters with those obtained by Star, CStar and ICSD for the same β. Table 3 shows the aforementioned comparison; in this table, columns “Gpr” and “Dty” represent the number of clusters and the density of those clusters respectively.

As it can be noticed from Table 3, ICSD outperforms Star and SComp in all datasets getting a less number of clusters with a higher density and it also obtains similar results than those obtained by Cstar.

[2] We selected 1000 because it is a number neither big nor small considering the size of *Reu-To*.

5 Conclusions

In this paper, a new algorithm called Incremental Clustering by Strength Decision (ICSD) has been proposed. ICSD algorithm builds a set of dense and overlapped clusters applying a new heuristic for covering a thresholded similarity graph which allows its application for incremental environments.

The heuristic introduced by ICSD algorithm processes only the clusters actually affected by additions, which makes ICSD to save time, making it able to efficiently manage multiple insertions in incremental environments.

ICSD algorithm was compared against other graph-based algorithms on six document collections. The experimental results show that our proposal outperforms those methods. Moreover, ICSD achieves better time performance than previous incremental overlapped graph-based algorithms in incremental datasets.

As future work we will develop a version of ICSD that allows additions and deletions, in order to increase its applicability in other environments.

References

1. Gil-García, R.J., Badía-Contelles, J.M., Pons-Porrata, A.: Extended Star Clustering Algorithm. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 480–487. Springer, Heidelberg (2003)
2. Pérez-Suárez, A., Medina-Pagola, J.E.: A Clustering Algorithm Based on Generalized Stars. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 248–262. Springer, Heidelberg (2007)
3. Gago-Alonso, A., Pérez-Suárez, A., Medina-Pagola, J.E.: ACONS: A New Algorithm for Clustering Documents. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 664–673. Springer, Heidelberg (2007)
4. Pérez-Suárez, A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Medina-Pagola, J.E.: A New Graph-Based Algorithm for Clustering Documents. In: ICDM-Workshops 2008, pp. 710–719 (2008)
5. Aslam, J., Pelekhov, E., Rus, D.: The star clustering algorithm for static and dynamic information organization. Journal of Graph Algorithms and Applications 8(1), 95–129 (2004)
6. Pons-Porrata, A., Ruiz-Shulcloper, J., Berlanga-Llavori, R., Santiesteban-Alganza, Y.: Un algoritmo incremental para la obtención de cubrimientos con datos mezclados. In: CIARP 2002, pp. 405–416 (2002)
7. Pons-Porrata, A., Berlanga-Llavori, R., Ruiz-Shulcloper, J.: On-line event and topic detection by using the compact sets clustering algorithm. Journal of Intelligent and Fuzzy Systems 12(3), 185–194 (2002)
8. Ruiz-Shulcloper, J., Sanchez, G., Abidi, M.A.: Clustering in mixed incomplete data. Heuristics and Optimization for Knowledge Discovery, 88–106 (2002)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons Inc., New York (2001)
10. Banerjee, A., Krumpelman, C., Basu, S., Mooney, R., Ghosh, J.: Model based overlapping clustering. In: KDD 2005, pp. 532–537 (2005)
11. Kuncheva, L., Hadjitodorov, S.: Using diversity in cluster ensembles. In: IEEE SMC 2004, The Netherlands, pp. 1214–1219 (2004)

The Multi-level Learning and Classification of Multi-class Parts-Based Representations of U.S. Marine Postures

Deborah Goshorn, Juan Wachs, and Mathias Kölsch*

MOVES Institute
Naval Postgraduate School
Monterey, CA, USA

Abstract. This paper primarily investigates the possibility of using multi-level learning of sparse parts-based representations of US Marine postures in an outside and often crowded environment for training exercises. To do so, the paper discusses two approaches to learning parts-based representations for each posture needed. The first approach uses a two-level learning method which consists of simple clustering of interest patches extracted from a set of training images for each posture, in addition to learning the nonparametric spatial frequency distribution of the clusters that represents one posture type. The second approach uses a two-level learning method which involves convolving interest patches with filters and in addition performing joint boosting on the spatial locations of the first level of learned parts in order to create a global set of parts that the various postures share in representation. Experimental results on video from actual US Marine training exercises are included.

1 Introduction

The ability to automate the evaluation of human performance in training exercises using computer vision and behavior analysis is of recent interest in several research fields. It is a complex goal, but a building block of this goal is to create computer vision algorithms to detect the atomic events seen in training exercises. This paper is a result of yielding the fundamental posture recognition computer vision algorithms to support the automation of evaluating US Marines in their training exercises.

The four fundamental postures of a US Marine in training are the four torso orientations as illustrated in Figure 1. Significant posture changes are detected in order to evaluate high-level behavior anlaysis of Marines in training[9]. To recognize the four *object types*, or postures, this paper investigates the parts-based object representations and the multiple levels of learning of the parts that represent each posture in order to obtain robust representations of postures.

* The authors greatly appreciate and acknowledge the invaluable contribution by Noah Lloyd-Edelman with the video dataset.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 505–512, 2009.

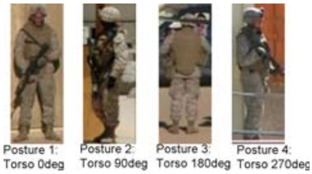


Fig. 1. Example training image from each object (US Marine posture) type

1.1 Related Work

This work is the first attempt to model multi-class human postures using parts-based approach on uniformed soldiers in a cluttered environment. It is a challenging problem, and this paper investigates the possibility of using multi-level learning of parts-based representation for multiclass body postures of uniformed soldiers to overcome the challenges of representing such diverse possible appearances of one posture class. This work is not the first attempt to represent objects as parts-based representations. In [1], parts representation of faces are learned using Multiple Cause Vector Quantization and Multiple Cause Factor Analysis, which are similar to Principle Component Analysis and Non-negative Matrix Factorization. In [3] and [4], a Bayesian approach to learning parts-based representations of objects is presented. The most related to this presented work is that of [2] and [5]. The second classifier presented is a variant of [2]. The first classifier presented in this paper is a multi-class version similar in learning, but different in classification to the pedestrian detector in [5].

2 Multi-level Learning and Classification of Parts-Based Object Representations

In order to use represent postures, or more generally, *object types*, with sparse representations, there must first be an attempt to learn the parts that make up the representation. It is advantageous to learn parts progressively, requiring multiple levels of learning. At each higher level of learning, the learning algorithm is more sophisticated, and the part learned is more sophisticated in its representation of the object type. For example, at the lowest level of learning, unsupervised learning like clustering learns salient features, or parts, from unlabeled parts data, that sparsely represents the object in common to the data. However, learning parts at this level is not always enough. The more levels of learning, the more sophisticated the parts-based representations are of the data.

The levels of learning may involve either purely one type of object or all types of objects. In the former, parts-based representations are learned for the sole purpose of object detection of that object type. There is no learning of parts that are discriminant between other object types. The second type of higher levels of learning which involve all object types, learn parts of object types that discriminate between object types. In this case, object types may even share parts in their individual parts-based representations. In this section, example parts-based recognition classifiers from both approaches are presented.

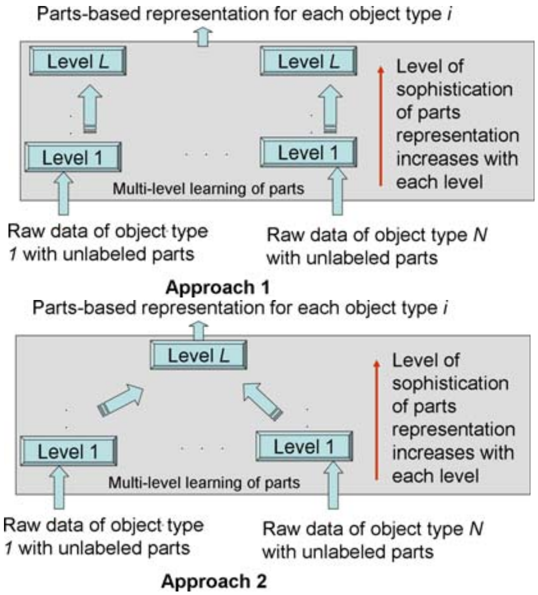

Parts-based representation for each object type *i*
Level *L*
Level *L*
Level 1
Level 1
Multi-level learning of parts
Level of sophistication of parts representation increases with each level
Raw data of object type 1 with unlabeled parts
Raw data of object type *N* with unlabeled parts
Approach 1
Parts-based representation for each object type *i*
Level *L*
Level 1
Level 1
Multi-level learning of parts
Level of sophistication of parts representation increases with each level
Raw data of object type 1 with unlabeled parts
Raw data of object type *N* with unlabeled parts
Approach 2

Fig. 2. Two approaches to multi-level learning of parts-based object representations

2.1 Approach 1 Example Parts-Based Object Recognition Classifier

One classifier presented in this paper is an example of a parts-based object recognition classifier that learns its parts in two levels, where all levels learn using data from one object type only. In other words, it is an instance of *Approach 1* with $L = 2$ from Figure 2. This classifier is similar to [5] in the learning part, but different in the classification process as soon described.

Level One Learning of Parts. Level one learning is entirely unsupervised with respect to learning the *parts* that represent an object type. For this classifier approach, the level one learning module as depicted in 3, inputs N_1 gray-scale training images of one object type 1, and outputs a *dictionary*, or set, of parts representing that particular object type. This learning level involves three majors steps. First, the Harris corner detector [8] is used to find the interest points on the object type attempted to be learned. Secondly, the interest patch, or window of size p x p extracted around each interest point, is collected from each training image. Here, p is fixed to 9, a mid-range value of standard patch sizes. Finally, a clustering algorithm, like the $K-$means clustering algorithm, is used to cluster the extracted patches (from all training images of a posture type) into K_1 clusters, where K_1 is the number of parts selected to comprise the dictionary for object type 1. The dictionary of *parts* is simply the resulting cluster means. In other words, if object part i is denoted as P_i, a p x p matrix; and the cluster i, denoted as C_i, is the set of all interest patches that fell into the i^{th} cluster, then we have the following: $P_i = \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} ip_j$, where ip_j is the jth interest patch that was clustered in cluster C_i and $|C_i|$ is the cardinality of i^{th} cluster.

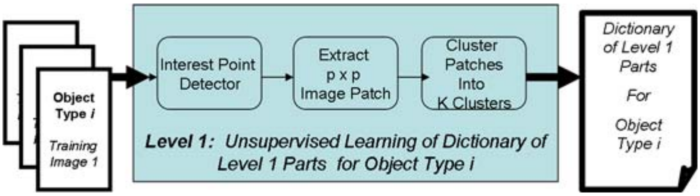

Object Type i
Training Image 1
Interest Point Detector
Extract p x p Image Patch
Cluster Patches Into K Clusters
Level 1: Unsupervised Learning of Dictionary of Level 1 Parts for Object Type i
Dictionary of Level 1 Parts For Object Type i

Fig. 3. Level 1 Learning of Parts

The distance metric used for clustering the image patches is the Normalized Grayscale Cross- Correlation, NGCC. If ip_j is the j^{th} interest patch and P_i is the i^{th} part (cluster center), then the NGCC between the interest patch and the cluster center is the following: $NGCC(ip_j, P_i) = \frac{\sigma^2_{ip_j,P_i}}{\sigma_{ip_j}\sigma_{P_i}}$. Where $\sigma^2_{ip_j,P_i}$ is the covariance between the interest patch and the cluster center, or equivalently, object part; and σ_{ip_j} and σ_{P_i} are the respective standard deviations of the interest patch and the object part.

This Level 1 learning module is extremely simple and fast as it is an unsupervised learning approach to learning the dictionary of parts. However, the number of parts chosen to represent an object type significantly affects the quality of the dictionary of parts. Careful consideration and several trials went into the final value of $K_i = 260$ parts for each object parts-based representation.

Level Two Learning of Parts. In the second level of learning, the object parts $P_1, P_2, \ldots P_{K_i}$ learned for representing object type i increase by one more level in sophistication of their meaning for representing the object type. For Approach 1, this is done by learning the nonparametric two-dimensional spatial frequency distribution for each part for an object types' parts-based representation.

The input of the level 2 learning module for the dictionary of parts for an object type is the set of training images for this particular object type. The output is the set of two-dimensional spatial frequency distribution estimators for each dictionary part of that object type. Thus there are K_i distribution estimators for the object type 1 dictionary of parts. As a result, each level 1 object part, $P_1, P_2, \ldots P_{K_i}$ has a nonparametric spatial distribution attached to it, thus producing a level 2 set of parts to represent object type i.

Classification Using Approach 1 Parts-based Representation. Classification of an object's type is similar to Level 2 learning module. It attempts to capture the level-2 parts that are in the test object and compare it with the parts of each object type learned a priori.

First, an interest point detector is applied to find the interest points. Second, patches of size p x p are extracted, called interest patches. Third, the NGCC distance is computed between each interest patch and each object part of *each* object type. Only those interest patches which yield an NGCC above a threshold pass to the next step. The next step is to see (1) which object type received the most *matches* with its parts and the interest patches, (2) which object type received *matches* whose interest patches had similar spatial distributions to those of the object parts, and (3) which object type received the highest distance. The object type which satisfies all three criteria is chosen.

2.2 Approach 2 Example Parts-Based Object Recognition Classifier

In this section, a parts-based classifier is presented that takes on the style of Approach 2 of multi-level learning of parts-based object representations. In the second approach to multi-level learning of parts-based representations, the highest level of learning, which is again $L = 2$, involves all possible object types in order to choose parts that better discriminate between object type, as portrayed in Figure 2. This approach is similar to that of [2] except that in our approach a multi-scale-class grouping algorithm was adopted to improve the recognition accuracy.

Level One Learning of Parts. As depicted in Figure 4, the level 1 learning module takes as input training images of a particular object type and outputs a set of level-1 parts that describe that object type. This approach is also unsupervised in the way that it does not have labeled parts already to work with. It discovers the parts itself. However, it does not use clustering to yield the level-1 parts, as in the example for Approach 1. This level 1 module happens to use convolution with several types of filters to create more than one representation for each interest patch extracted from an object in the training image. More specifically, it applie8s 2D convolution with four filters: a delta function, x and y derivatives and a Gaussian, see Figure 5 and the following equation: $v_i(x, y) = [(I * f_i) \otimes P_i] * l_x^T l_y$, where $*$ is the convolution operator, $\otimes$ is the normalized cross correlation operator, $v_i(x, y)$ is the feature vector entry i, f is a filter, P is a patch, and l_x and l_y are the x, y location vectors with respect to the center of the image respectively. The battery of filters f used are depicted in Figure 5. The patches' sizes are selected randomly between 9x9 to 25x25 since it showed better results than using a fixed patch sizes. The location information for the interest patch is recorded as well as the four responses, or level one parts, from the filtering. Location is stored in two Gaussian 1D vectors, where each has an offset equal to the x and y distances, respectively. This distance is computed in a constant time. Note, each level-1 part, P_i, has one location and four responses associated to it.

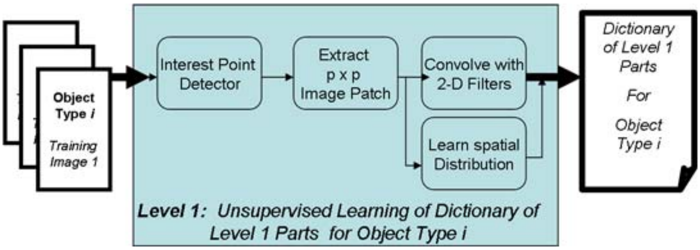

Object Type i
Training Image 1
Interest Point Detector
Extract p x p Image Patch
Convolve with 2-D Filters
Learn spatial Distribution
Level 1: Unsupervised Learning of Dictionary of Level 1 Parts for Object Type i
Dictionary of Level 1 Parts For Object Type i

Fig. 4. Level 1 Learning of Parts

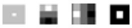


Fig. 5. A delta function, x and y derivatives and a Gaussian

Level Two Learning of Parts. In the second level of learning parts, the learning module takes as input the set of level 1 parts learned from *each* object type and outputs a new set of level 2 parts for *each* object type that attempted to maximize inter-class similarities. A joint boosting algorithm is used for multi-class detection and classification, see [2]. This is based on a boosting algorithm where weak learners are sequentially added to form a strong classifier. For each class, a strong learner $H(P_i, c)$ is computed. Where P_i is the level 1 part and c is the object type/class. Each round of boosting, a search is conducted on all the components, f of the level 1 part P_i, for each component, search over all the discrete values of possible thresholds Θ and for each couple f, Θ, find the optimal regression parameters a_S and b_S. Finally, select f, Θ, a_S, b_S that minimizes a cost function. Note, that this level of learning, unlike the previous approach, spans all possible object types/classes in order to choose shared features/parts that attempt to optimize overall object type recognition accuracy.

3 Experimental Results

There were four experiments conducted on a restricted access video sequence of 169 frames called "MOV007_seq1" which used a pan-tilt-zoom camera to record a dynamic field of view including three patrolling Marines that interact closely in distance with each other. Results from one particular video frame is shown in Figure 6. This sequence is part of a collection of clips showing US Marine's outside training exercises as part of a project on behavioral analysis [9]. A dataset including annotated still images from multiple marines poses was used to train the classifiers.

First, the Approach 1 multi-level learned parts-based classifier currently executes in a single scale, so there needs to be a standard person detector first. Thus, the first scenario/experiment is that of running the Felzenszwalb person detector [7] first. Then, the resulting bounding box of the detected person is resampled to a standard size which then inputs to the Approach 1 type posture classifier, which attempts to match the parts of the detected person to the set of parts of each posture learned. As described earlier, the most likely posture

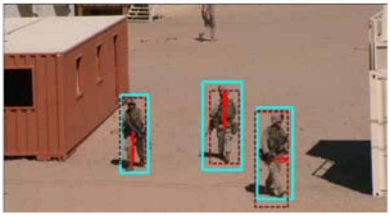


Fig. 6. A video frame with annotated detections/postures

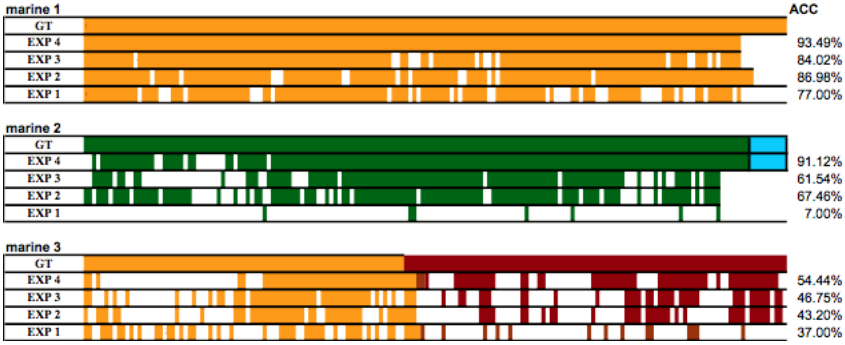


Fig. 7. Temporal analysis and classifier accuracy of all four experiments (GT=ground truth, then Exp. 4, Third row is Exp 3, and fourth row is Exp. 2. Colors: or- 0deg;b-90deg; gr-180deg;r-270deg;blnk-not detected/confused).

is outputted. Note, the detection time is 6-7 sec. The Approach 1 recognition takes 5-6 seconds. The second experiment is similar to the first, except instead of executing the Approach 1 classifier, the Approach 2 classifier, single scale is executed. The detection time is 7-8 seconds and recognition time is 1-2 seconds. The third experiment is similar to the first two, however the Approach 2 example classifier is executed at multi-scale. The detection time in this case is 6 -7 seconds, while the recognition time is 6- 7 sec. Finally, the fourth experiment, the Approach 2 classifier stands on its own and performs both detection and multi-scale posture recognition. This takes 272- 273 seconds.

Figure 7 displays a temporal analysis of the accuracy of the experiments. Going through the video frames, comparing the actual ground truth with the results produced very useful inferences regarding multi-level learning of parts-based recognitions: (1) Since the multi-level learned parts-based classifier using the single-scale Approach 2 learning layout, outperforms the Approach 1 multi-level learned parts-based classifier, it is better it to learn multi-level learned parts which are produced progressively *and* simultaneously for all object parts in a manner to select and share level 2 parts which discriminates between object types' parts; (2) The multi-scale version of the Approach 2 classifier outperforms the single-scale version of the Approach 2 classifier, inferring that resizing the detected persons reduces recognition accuracy.

Finally, for a large period of the video sequence, Marine 2 and Marine 3 are overlapping, so Marine 2 was not detected often because the detector creates only one large detected bounding box for both Marines. Also, when a Marine walking torso 0 degrees has his head turned sideways, the posture recognition classifier gets confused. Finally, Marine 2 was walking 180 degrees (away from camera) most of the time; however Approach 1 classifier classified him mostly as walking 0 degrees (toward camera). (3) From these latter statements, one can infer that with the addition of a head detector and head orientation classifier, the latter three problems would be mitigated. The head detector and head orientation recognition can be thought of as a third level in learning parts for postures, since it is more sophisticated part than the level 1, and level 2 parts proposed in this paper. In conclusion, future work entails the addition of a head detector with head orientation, a level 3 learned part, to the Approach 2 of parts-based representation of US Marine postures.

References

1. Ross, D.A., Zemel, R.S.: Learning Parts-Based Representations of Data. J. Mach. Learn. Research 7, 2369–2397 (2006)
2. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. In: IEEE PAMI (2007)
3. Fergus, R., Perona, P., Zisserman, A.: Object recognition by unsupervised scaleinvariant learning. In: CVPR 2003 (2003)
4. FeiFei, L., Fergus, R., Perona, P.: OneShot learning of object categories. In: PAMI 2006 (2006)
5. Leibe, B., Leonardis, A., Schiele, B.: Robust Object Detection with Interleaved Categorization and Segmentation. In: IJCV (2007)
6. Wachs, J.P., Goshorn, D., Kölsch, M.: Recognizing Human Postures and Poses in Monocular Still Images. In: IPCV 2009 (2009)
7. Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: CVPR 2008 (2008)
8. Harris, C., Stephens, M.J.: A combined corner and edge detector. In: AVC (1988)
9. BASE-IT: Behavior Analysis and Synthesis for Intelligent Training, http://www.movesinstitute.org/base-it/index.html

The Representation of Chemical Spectral Data for Classification

Diana Porro[1,2], Robert W. Duin[2], Isneri Talavera[1], and Noslen Hdez[1]

[1] Advanced Technologies Application Centre, Cuba
[2] Pattern Recognition Group, TU Delft, The Netherlands
{dporro,italavera,nhernandez}@cenatav.co.cu, r.duin@ieee.org

Abstract. The classification of unknown samples is among the most common problems found in chemometrics. For this purpose, a proper representation of the data is very important. Nowadays, chemical spectral data are analyzed as vectors of discretized data where the variables have not connection, and other aspects of their functional nature e.g. shape differences (structural), are also ignored. In this paper, we study some advanced representations for chemical spectral datasets, and for that we make a comparison of the classification results of 4 datasets by using their traditional representation and two other: Functional Data Analysis and Dissimilarity Representation. These approaches allow taking into account the information that is missing in the traditional representation, thus better classification results can be achieved. Some suggestions are made about the more suitable dissimilarity measures to use for chemical spectral data.

Keywords: Pattern Recognition, Chemometrics, Classification, Spectral Data, Dissimilarity Representation, Functional Data Analysis.

1 Introduction

One of the main problems that can be found in any research area is related to the classification of unknown objects. A good representation of the data is one of the most important aspects to be considered in this process. The more information about the real data is described in its representation, the higher the probability of a good classification of the samples.

Although chemical spectral data are typically curves plotted as functions of wavelengths, product concentration, etc., they are traditionally represented as a sequence of individual observations (features) made on the objects, ignoring important aspects of their functional nature i.e. connectivity, shape changes, etc.

Functional Data Analysis (FDA) [1] and Dissimilarity Representation (DR) [2] are rather new approaches that, in their own way, can take the functional information into the data representation. FDA is an extension of the traditional multivariate analysis for data with a functional nature, and is based on considering the observed spectra as a continuous real-valued function instead of an array of individual observations. Several classical multivariate statistical methods been extended to work on it e.g. linear discriminant analysis (LDA) [3]. In the case of linear modeling, studies have also

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 513–520, 2009.

been made in regression [4]. A number of estimation methods for functional nonparametric classification and regression models have been introduced. Namely, k-Nearest Neighbor classifier (k-NN) [5], kernel classifiers e.g. Support Vector Machine (SVM) based on the Radial Basis Function (RBF) methods [6], [7], showing its application for chemical spectral data.

Although profound studies of the DR on chemical spectral data sets have not been done, there are already some results on spectral data in general [8], demonstrating its advantages for its classification. In this approach, based on the important role that proximities play in the classification process, the authors propose to work on a space defined by the dissimilarities between the objects [2]. This way, the geometry and the structure of a class are defined by the dissimilarity measure, by which we can take into account the information that can help to discriminate between objects of the different classes. So, the selection of a suitable measure for the particular problem is important. The DR has shown to be advantageous in problems where the number of objects is small, and also when they are represented in high dimensionality spaces, which are both common characteristics of chemical spectral data sets.

On the chemometrics side, some work has been done in the comparison of chemical spectral data. In [9], the authors are looking for similarity measures for infrared (IR) spectrometry. A more recent research [10] is about the comparison of drugs UltraViolet (UV) spectra by clustering, where they also try different dissimilarity measures.

The goal of this paper is to show, how the classification results can improve by using representations of the data that give more information about the real spectra than the feature representation. With this purpose, we make a comparison of the performance of 1-NN, Regularized LDA (RLDA), Soft Independent Modeling of Class Analogy (SIMCA) [11] and SVM classifiers on the three mentioned representations: feature, FDA and DR of four chemical spectral datasets. We also make a study of some dissimilarity measures that have already been used on these types of data, in order to propose which could be more suitable to take into account the main differences that can exist in spectral data sets: structure (shape) and/or concentration or intensity.

2 Functional Data Analysis

Functional Data Analysis (FDA) [1] was proposed as a way to retrieve the intrinsic characteristics of the underlying function from the discrete functional data. In this approach, the observations can be seen as continuous single entities, instead of sets of different variables. However, if the algorithms work on the functional spaces, their infinite dimensions can lead to theoretical and practical difficulties. To deal with the infinite dimensional problem, a filtering approach was constructed to reach a representation of a finite dimensionality.

For this approach, we have to select a proper family of basis functions to match the underlying function (s) to be estimated. In the case of spectral data, the basis of B-splines seems to be the most appropriate. A number of knots (points) between the start and end wavelengths has to be chosen, and a B-spline is run from one knot to another; the different splines overlap. The spectral function $x_i = x_i(\lambda)$ for sample i and wavelengths λ , can be described by the linear combination of the basis functions $x_i = \sum_{k=1}^{K} c_{ik}\phi_k$, where $\{\phi_k\}_{k=1}^{K}$ is the basis of B-splines with K the number of basis functions, and c_{ik} the B-spline weights (coefficients). These are computed by minimizing the vertical distance between the observed spectral information and the fitted curve:

$$\min_{c_{ik}} \sum_{j=1}^{m} (x_{ij} - \sum_{k=1}^{K} c_{ik}\phi_k(\lambda_j))^2 ,$$

where x_{ij} is an element of the matrix conformed by a set of i spectra of j wavelengths. The function will be explained by the coefficients and the methods will take these as the new representation of the data instead of the original data points.

3 Dissimilarity Representation

The Dissimilarity Representation (DR) [2] proposes to work on the space of the proximities between the objects, instead of the space defined by their characteristics (features), as it is usually done.

In the new representation, instead of having a matrix $\mathrm{X}(m \times n)$, where m goes for the objects (spectrum) and n for the measured variables e.g. wavelengths, the set of objects will be represented by the matrix $\mathrm{D}(m \times q)$. This matrix contains the dissimilarity values between each object $x \in \mathrm{X}$ and the objects of the representation set $\mathrm{R}(p_1, p_2, ..., p_q)$, $d(x_m, p_q)$. The elements of R are called prototypes, and have preferably to be selected by some prototype selection method [12]. These prototypes are usually the most representative objects of each class $(\mathrm{R} \subseteq \mathrm{X})$, but the whole set of objects X can be used too, obtaining the square dissimilarity matrix, $\mathrm{D}(m \times m)$; R can also be a completely different set of objects.

For the DR three main approaches exist. In the first, the given dissimilarities are addressed directly e.g. k-NN. Another one is based on an approximate embedding of the dissimilarities into a pseudo-Euclidean space. The third and last one is defined as the dissimilarity space $\mathscr{D} \subseteq \mathbb{R}^{\mathrm{n}}$, which is the one to be used here. This space is generated by the column vectors of the dissimilarity matrix, where each dimension corresponds to the dissimilarity value between the objects and a prototype $d(\cdot, p_q)$.

As the dissimilarities are computed to the representation set, already a dimensionality reduction is reached and therefore it can be less computationally expensive for the classification process. Furthermore, any traditional classifier that operates on feature spaces can also be used in the dissimilarity space.

3.1 Dissimilarity Measures

A general dissimilarity measure for all types of data does not exist. For each problem at hand, a dissimilarity measure adapted to the type of data should be selected. In the case of spectral data, the connectivity i.e. continuity, ordering between the measured points, may be taken into account. In this work, we present some initial studies on dissimilarity measures for the dissimilarity representation of chemical spectral data, based on: their structures (shape changes) and/or concentration or intensity changes.

For this purpose, we studied dissimilarity measures that are more commonly used in the comparison of chemical spectral data (see Section 1). Such is the case of the very well known Manhattan (L1-norm) and Euclidean distances.

In [13], the Spectral Angle Mapper (SAM) measure (Eq. 1) was proposed for spectral data. If we have samples (spectra) $x_1, x_2 \in \mathbb{R}^n$, the SAM dissimilarity is computed as follows:

$$d(x_1, x_2)_{sam} = \mathrm{ar}\cos\left(\sum_{j=1}^{n} x_{1j} x_{2j} \Big/ \sqrt{\sum_{j=1}^{n} x_{1j}^2 \sum_{j=1}^{n} x_{2j}^2} \right). \tag{1}$$

$$d(x_1, x_2)_p = 1 - \left(\sum_{j=1}^{n} \left(x_{1j} - \bar{x}_1\right)\left(x_{2j} - \bar{x}_2\right) \Big/ \sqrt{\sum_{j=1}^{n} \left(x_{1j} - \bar{x}_1\right)^2 \sum_{j=1}^{n} \left(x_{2j} - \bar{x}_2\right)^2} \right). \tag{2}$$

The dissimilarity measure in Eq. 2 is based on the Pearson Correlation Coefficient (PCC), and measures the angle between two vectors, like the SAM measure. The PCC can be also seen as the cosine of the angle between two mean-centered samples. Although the previous dissimilarities are of the most used measures in the comparisons of chemical spectral data, the connectivity between the n measured variables is not taken into account in neither of them. The variables could be easily reordered and the same dissimilarity value is obtained.

The Kolmogorov-Smirnov distance (KS) (Eq. 3) is a dissimilarity measure between two probability distributions:

$$d(x_1, x_2)_{ks} = \max_j \left(\left| \hat{x}_{1j} - \hat{x}_{2j} \right| \right). \tag{3}$$

$\hat{x}_{1j}$ and $\hat{x}_{2j}$ are the cumulative distribution functions of the object vectors. Spectra need to be normalized to unit area, thus the areas under the original distribution of the data can be compared and their shape reflected.

In [8], the authors propose to compute the Manhattan measure on the first Gaussian derivatives (Eq. 4) of the curves (Shape measure), to take into account the shape information that can be obtained from the derivatives:

$$d(x_1, x_2)_{shape} = \sum_{j=1}^{n} \left| x^{\sigma}{}_{1j} - x^{\sigma}{}_{2j} \right| \quad \text{with} \quad x^{\sigma} = \frac{d}{dj} G\left(j, \sigma\right) * x. \tag{4}$$

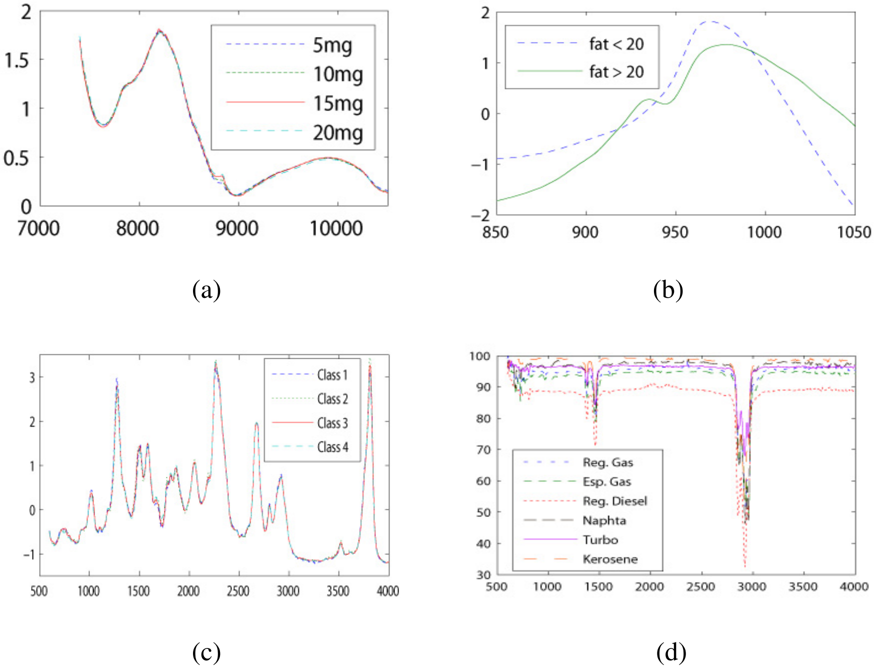
where $*$ denotes convolution and σ stands for a smoothing parameter.

4 Experimental Section and Discussion

To evaluate the performance of different classifiers, a comparative study will be made with the three different representations of the data and four classifiers: 1-NN, RLDA, SIMCA and SVM. All the experiments were performed in Matlab. For FDA the FDAFuns toolbox was used, and the PRTools toolbox for the DR and classification of the data. For FDA, each spectrum was represented by an l order B-spline approximation, with K basis functions. The optimal values for the number of B-spline coefficients and the degree of the spline was chosen using leave-one-out cross validation. For the DR, all the samples were used as representation set.

The comparison among the models was made by the averaged error of a 10 times 10-fold cross-validation (CV), on the three representations: feature, functional (FDA), and the DR for the different dissimilarity measures presented in Section 2. For the SVM classifier, after trying with different kernels, the best results were achieved with the Gaussian kernel for Tecator dataset and the linear kernel for the rest. The regularization parameter C was optimized, as well as the number of principal components in SIMCA. To find the regularization parameters of RLDA an automatic regularization process was done. The details of all datasets are related in Table 1.

The first data set (Fig. 1a) is composed by near infrared (NIR) transmittance spectra of pharmaceutical tablets [14] of four different (classes) dosages of nominal content of active substance. In this data, the spectra of the samples of the different classes



(a)

(b)

(c)

(d)

Fig. 1. Spectrum of one sample from each of the classes of each datasets: a) Tablet, b) Tecator, c) Oil and d) Fuel

are very similar, they variate in the intensity of only one peak at 8830 cm^{-1}. This peak corresponds to the only visually characteristic band of the active substance. Multiplicative scatter correction (MSC) was used as preprocessing method.

The second, named Tecator [15] (Fig. 1b), consists of NIR absorbance spectra of meat samples. In this data, the samples of the two classes differ in their fat content which is reflected in changes in the shape of the spectra (structure). Standard Normal Variate (SNV) was used as preprocessing method. The second derivative of the spectra is computed on the functional representation.

The third dataset consists of oil samples of different origins, analyzed by Mid-Infrared (MIR) technique [16] and was transformed to have zero mean and unit variance. The variations in the spectra of the classes are based in the difference in concentration of some substances and some shape changes also exist.

And the last dataset consists of fuel samples of Fourier Transform Infrared (FT-IR) transmittance spectra; base line correction and smoothing were performed on the data. The samples of these classes differ in the substances by which they are composed (structure), and therefore they differ in shape.

Table 1. Details about the # samples, features and samples per class of each dataset. The last column is related to the # basis functions used for the FDA of each dataset.

DataSet	#Samples	#Features	# Samples per Class	#Basis Functions
Tablets	310	404 (7400 to 10500 cm^{-1})	Types: A (5mg), B (10mg), C(15mg) and D(20mg)	100
Tecator	215	100 (850-1050 nm)	Fat content: Low (77) , High (138)	48
Oils	80	571 (600-4000 cm^{-1})	Origin: A (18), BB (8), BC (29) and D (25)	100
Fuels	80	3528 (600-4000 cm^{-1})	Type: Regular Gasoline (16), Especial Gasoline (15), Regular Diesel (16), Naphtha(16), Turbo Diesel(8) and Kerosene(9)	300

As can be seen in Table 2, in general for the four datasets, the SVM shows good results on all the representations, outperforming the rest of the classifiers. These could be due to these datasets are mostly non-linear. The exception is Tablets, where RLDA seems to outperform the other classifiers for its feature and functional representation, but in the DR, SVM again shows superiority. The experiments show that, most of the time, for most classifiers, their accuracy improves when using the DR and functional representation of these datasets. This demonstrates the importance of a good and descriptive representation of the data. In the case of DR, the results depend on whether a suitable dissimilarity measure is used to explain the discriminative characteristics of the curve, in order to obtain a better and more reliable classification of the data. It is worth to notice that, for both representations, the dimensionality of the datasets are reduced to half (or more) of the dimensionality of the feature representation. From the comparison of the different dissimilarity measures used, we can observe that very good results are achieved with the Shape dissimilarity, in which connectivity and shape information are considered. This proves the fact outlined in the previous paragraph, and suggests that this dissimilarity measure could be a good option for our purpose.

If we compare the results with the functional representation (FDA) and the DR of the data, they show that both approaches are good when the shape variations between the samples of different classes are appreciable. But it can be observed that, the DR gives the best results for most datasets (with the Shape measure). It shows the capability of the Shape measure, which performs well not only in datasets where the differences are based in changes in the curvature of the spectra, but also when concentration or intensity changes are present. On the other hand, in datasets like Tablet, where the functional information to be extracted is very poor, the FDA does not work very well. This lack of information in the functional data, can also be due to some of the information could have been lost by using only the coefficients resulting from the projection of the function in the B-spline basis.

Table 2. Averaged CV error with its standard deviation (%). The results are shown for the four classifiers on the feature, functional, and DR of each dataset for the six dissimilarity measures presented. The numbers highlited in bold and underlined, stand for the lowest error among all the representations for each classifier. In the case of the dissimilarities, the one that performs best in general for each dataset is also highlited in italic.

Data Sets		Feature	FDA	Dm	De	Dsam	Dpcc	Dks	Dshape
Tablets	1-NN	12,9(0,18)	**9(0,15)**	48,2(0,03)	13(0,02)	25,1(0,01)	13(0,02)	14,5(0,01)	***15,7(0,06)***
	RLDA	9,9(0,06)	10,6(0,09)	6,8(0,02)	11(0,1 e^{-17})	15,8(0,01)	8,4(0,03)	30,3(0)	***5,1(0,1 e^{-17})***
	SIMCA	25,7(0,16)	23,3(0,27)	17,2(0,02)	16(0,03)	20,2(0,06)	35,4(0,03)	26,5(0,02)	***10,7(0,03)***
	SVM	13,6(0,03)	16(0,09)	5,1(0,01)	5,3(0,03)	6,8(0,1 e^{-17})	14,8(0,02)	14,1(0,02)	***5,1(0,01)***
Tecator	1-NN	3(0,17)	2,2(0,17)	5,3(0,14)	5,3(0,19)	***1,9***(0,04)	11,2(0,04)	11,1(0,04)	3,3(0,04)
	RLDA	4,7(0,02)	3,5(0,2 e^{-17})	4,7(0,09)	4,7(0,09)	1,4(0,19)	3,8(0)	15,6(0,19)	***1,4*****(0,04)**
	SIMCA	2,5(0,12)	**2**(0,2)	9,4(0,09)	9,8(0,4 e^{-17})	***2,4***(0,9 e^{-17})	16,8(0,9)	15,3(0,9)	3,2(0,04)
	SVM	1,9(0)	**1**(0)	1(0,04)	2,8(0,2 e^{-17})	***1***(0,2 e^{-17})	1,9(0,04)	4,7(0,2)	1,4(0,1 e^{-17})
Oils	1-NN	13,8(0,32)	**7,5(0,19)**	11,1(0,51)	13,1(0,47)	7,4(0,44)	13,1(0,47)	17,4(0,29)	***9,4(0,47)***
	RLDA	22,4(0,13)	20(0,4 e^{-17})	22,8(0,25)	21,4(0,12)	22,6(0,13)	23,6(0,12)	19(0,25)	***18,6(0)***
	SIMCA	7,9(0,56)	**6,6(0,62)**	16,3(0,81)	15,6(0,43)	17,9(0,42)	17(0,46)	19,2(0,62)	***14(0,36)***
	SVM	6,3(0)	**2,5(0)**	13,8(0,2)	15,9(0,37)	8,9(0,13)	8,8(0,4)	19,8(0,12)	***6,3(0)***
Fuel	1-NN	35,1(2,08)	17,7(1,71)	9,5(0,62)	33,3(0,75)	20,1(0,54)	14(0,52)	30,2(0,58)	***8,6(0,42)***
	RLDA	22,5(0)	21(0,79)	15,1(0,54)	39,8(1,16)	15,5(0,86)	19,6(0,42)	43,1(1,02)	***16,9(0,75)***
	SIMCA	30,4(3,73)	12,4(1,61)	12(0,38)	40,5(0,82)	20,4(0,65)	20(0,91)	57,5(0,49)	***11,9(0,43)***
	SVM	10(0,04)	7,5(0,4 e^{-17})	8,6(0,12)	25,3(0,25)	13(0,50)	16(0,25)	35,1(0,13)	***5,5(0,50)***

In the case of Tecator dataset, good results are achieved either with the FDA representation or the DR (for the different classifiers); there is barely a difference between the errors committed for some classifiers when operating on them (looking also at the standard deviation error). Nevertheless, FDA performed better in general. It can be explained by the fact that, from the functional point of view, a lot of information can be obtained when shape changes are present in the curve. So the FDA by B-splines is capable of using this information and the use of the second derivatives afterwards emphasizes the peaks in the curve, making easier to see the differences. In the Fuel dataset, a similar result could be expected if the same procedure is carried.

However, in spite of the good performance of the DR for most cases, this is not the case for Oil dataset. This suggests that, although the dissimilarity measures have shown their ability to discriminate between spectra that are very similar (see Tablet dataset in Fig. 1a); they might not be robust enough for cases like this, where the shape varies so abruptly and so frequently in the spectrum. Still, the results could be improved if the DR is computed on the FDA representation. Further researches most be done on this aspect.

5 Conclusions

We presented two alternative ways to improve the representation of chemical spectral data. The first makes use of the spectral connectivity by approximating the spectra by spline functions (FDA). The second makes use of the physical knowledge of the spectral background of the data by modeling their relations in a dissimilarity representation. Comparisons were made by classifying four chemical spectral datasets, expressed by their feature and the two other representations. It was shown that, with the studied representations, improved classification results can be obtained. But it shows that the use of either of them will depend on the characteristics of the data. We can also conclude that, for the comparison of spectral chemical data by their dissimilarities, the better results are obtained with measures that take the connectivity between the points, and shape information into account.

References

1. Ramsay, J.O., Silverman, B.W.: Functional Data Analysis, New York (1997)
2. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation For Pattern Recognition. Foundations and Applications 64 (2005)
3. Cardot, H., Ferraty, F., Sarda, P.: Functional linear model. Statist. Probab. Lett. 45, 11–22 (1999)
4. Preda, C., Saporta, G.: PLS regression on stochastic processes. Comput. Statist. Data Anal. 48, 149–158 (2005)
5. Cérou, F., Guyader, A.: Nearest neighbor classification in infinite dimension. ESAIM: Probability and Statistics 10, 340–350 (2006)
6. Villa, N., Rossi, F.: Support Vector Machine For Functional Data Classification. In: ESANN 2005 (2005)
7. Hernández, N., Biscay, R.J., Talavera, I.: Support Vector Regression Methods for Functional Data. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 564–573. Springer, Heidelberg (2007)
8. Paclik, P., Duin, R.P.W.: Classifying spectral data using relational representation. In: Spectral Imaging Workshop, Graz, Austria (2003)
9. Varmuza, K., Karlovits, M., Demuth, W.: Spectral similarity versus structural similarity: infrared spectroscopy. Anal. Chimica Acta 490, 313–324 (2003)
10. Komsta, L., Skibinski, R., Grech-Baran, M., Galaszkiewicz, A.: Multivariate comparison of drugs UV spectra by hierarchical cluster analysis-comparison of different dissimilarity functions. In: Annales Universitaits Marie Curie-Sklodowska, Lublin, Polonia, vol. 20, pp. 2–13 (2007)
11. Wold, S.: Chemometrics: Theory and Application. In: Kowalski, B.R. (ed.) ACS Symposium, vol. 52, pp. 243–282 (1977)
12. Yuhas, R.H., Goetz, A.F.H., Boardman, J.W.: Discrimination among semiarid landscape end members using the spectral angle mapper (SAM) algorithm. In: Third Annual JPL Airborne Geoscience Workshop, Pasadena, CA, pp. 147–149 (1992)
13. Pekalska, E., Duin, R.P.W.: Prototype selection for finding efficient representations of dissimilarity data. In: Kasturi, R., Laurendeau, D., Suen, C. (eds.) International Conference on Pattern Recognition, Quebec, Canada, vol. 3, pp. 37–40 (2002)
14. Tablets dataset, `http://www.models.kvl.dk/research/data`
15. Tecator dataset, `http://lib.stat.cmu.edu/datasets/tecator`
16. Oil dataset, `http://cac2008.teledetection.fr/shootout`

Visual Pattern Analysis in Histopathology Images Using Bag of Features

Angel Cruz-Roa, Juan C. Caicedo, and Fabio A. González

Bioingenium Research Group
Universidad Nacional de Colombia
{aacruzr,jccaicedoru,fagonzalezo}@unal.edu.co

Abstract. This paper presents a framework to analyse visual patterns in a collection of medical images in a two stage procedure. First, a set of representative visual patterns from the image collection is obtained by constructing a visual-word dictionary under a bag-of-features approach. Second, an analysis of the relationships between visual patterns and semantic concepts in the image collection is performed. The most important visual patterns for each semantic concept are identified using correlation analysis. A matrix visualization of the structure and organization of the image collection is generated using a cluster analysis. The experimental evaluation was conducted on a histopathology image collection and results showed clear relationships between visual patterns and semantic concepts, that in addition, are of easy interpretation and understanding.

1 Introduction

Medical research centers and medical schools today are facing the problem of analyzing huge volumes of images from ongoing studies and the normal clinical operation [7]. The amount of available visual information in medicine constantly grows and discovering visual patterns in a large collection of images is a challenging task. Currently, academic image collections for classroom study or advanced research in medicine are managed by an expert who carefully organize images according to domain knowledge criteria. However, these collections have no more than a few hundred images, since the capacity of human beings to deal with large data collections is limited. Computers are an important asset to support tasks such as the analysis of image structure [5] and the identification of common and distinctive visual patterns in large image collections[6].

A large collection of medical images may be organized according to several categories that describe anatomical or pathological properties, using metadata from a hospital information system or records from a medical research survey. So, given such a collection, the main goal is the characterization of those visual properties that are common to a set of semantically related images. In the context of this paper, this problem is denoted visual pattern analysis on an image collection. The identification of visual patterns on a collection of medical images may lead to a better understanding of biological structures and also to

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 521–528, 2009.

design computer aided diagnosis tools or educational applications to train new physicians [4]. Two main questions arise when dealing with the visual pattern analysis task: how does the system detect or identify patterns that compose image structures in the collection?, and how do those visual patterns relate with pathological concepts?.

In this paper we propose a framework to answer these two questions. First, to identify visual patterns inside an image collection, the use of a bag-of-features representation is proposed, in which a dictionary or codebook is defined by grouping features extracted from all individual images. This dictionary constitutes a representative set of the visual patterns in the image collection, that can be visually understood and interpreted by domain experts, a task that is not always possible using other variety of image representations. Second, the relationships between visual patterns and semantic concepts is analysed applying two complementary strategies: a correlation analysis and a cluster analysis. The correlation analysis allows to identify a set of visual patterns that are frequently associated with particular concepts, while the cluster analysis allows to visualize the distribution of patterns for similar images and the image collection structure. This framework has been applied to a collection of histopathology images showing how both, the feature dictionary and the subsequent analysis, are revealing the visual and semantic structure of the collection.

The bag-of-features representation has been successfully applied for classification of natural scenes [2] and medical images [6], but its applicability on histopathology images has been largely unexplored [1]. This paper also aims to evaluate the suitability of this approach for histopathology images under the proposed framework. The structure of this paper is as follows: Section 2 presents details of the bag-of-features approach. Section 3 discusses the identification of semantic relationships using correlation analysis and cluster analysis. Section 4 presents the experimental results on a histopathology image collection and finally Section 5 presents the conclusions and future work.

2 The Bag-of-Features Representation

The *bag-of-features* representation is an adaptation of the *bag-of-words* scheme used for text categorization and text retrieval. The key idea is the construction of a *codebook*, that is, a visual vocabulary, in which the most representative patterns are codified as *codewords* or visual words. Then, the image representation is generated through a simple frequency analysis of each *codeword* inside the image. This representation has been successfully applied in different image classification tasks. There are three main steps to build a *bag-of-features* representation [2]: (i) feature detection and description; (ii) codebook generation; and, finally; (iii) the *bag-of-features* construction. Figure 1 shows an overview of those steps. The *bag-of-features* approach is a novel and simple method to represent image contents using collection dependent patterns.

In this work the following strategy has been used to generate the bag of features representation for histopathology images: for feature detection, raw blocks

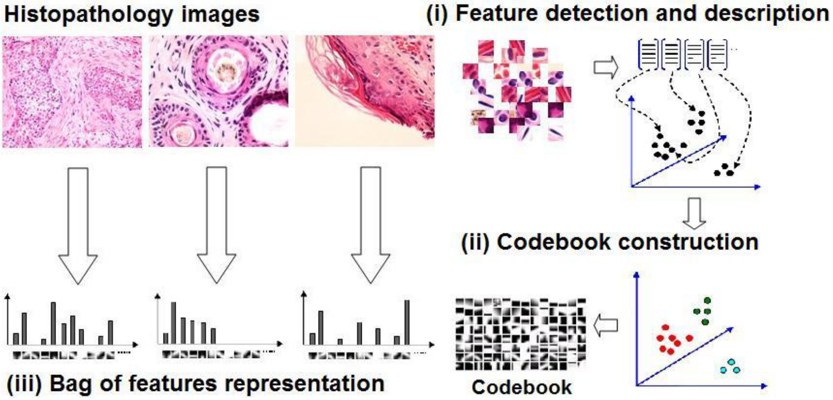

Histopathology images
(i) Feature detection and description
(ii) Codebook construction
(iii) Bag of features representation
Codebook

Fig. 1. Overview of the Bag of Features representation

are extracted from a regular grid on each image using 8 × 8 pixels per block. Each block is represented by the array of 64 gray level values which is used as feature vector. For codebook generation, the k-means algorithm is applied over the whole set of blocks. The size of the codebook, k, is an important parameter. It is expected that a moderately codebook size, in the order of hundreds, wold be enough to capture the most important patterns in the collection [1]. For the experimentation carried on in the present work, $k = 50$ was used based on previous findings in the same image collection [1]. Finally, the bag of features for each image is generated by counting the occurrence of visual words in the codebook.

3 Visual Pattern Analysis

The bag-of-features codebook constitutes a summary of the visual patterns present in the histopathology image collection. The hypothesis is that some of these visual patterns are related to histopathology concepts. In order to corroborate it, two strategies are applied, a correlation analysis and a cluster analysis.

3.1 Correlation Analysis

The goal of the correlation analysis is to measure the strength of the relationship between a particular visual pattern from the dictionary and a semantic concept. Images in the collection are known to be in one or several predefined categories or semantic classes. Then, we assume two random variables to analyse the correlation between them: semantic concepts and visual patterns. For semantic concepts the random variable is binary and indicates the presence or absence of the concept in the image. For visual words, the random variable is assumed continuous and corresponds to the relative frequency of the visual word in the image.

Following these assumptions, we can evaluate the correlation of visual patterns and semantic concepts. When a particular concept and a visual pattern are constantly exhibited in an image set, it is expected that the correlation between them has a positive value. On the other hand, if the visual pattern is not usually in those images that exhibit the concept, then a negative correlation is expected. Hence, the correlation analysis is useful to identify the set of most representative visual patterns associated to semantic concepts.

3.2 Clustering Analysis

A natural basis for organizing visual patterns is to group together those that share similar occurrence in images. The purpose of this cluster analysis is to generate a reordering of visual patterns to analyse the relationships with semantic concepts. Due to the large amount of images in a collection and also to a potential large dictionary of visual patterns, it is difficult to assimilate underlying relationships. Therefore, we follow a visual representation that is usually applied in bioinformatics to visualize and explore gene expression data in an intuitive manner for biologists [3]. We combine clustering methods with a graphical representation of the visual patterns in images by representing each occurrence value using a color in a matrix, as it is shown in Figure 4. A blue color indicates a low frequency of visual patterns in images, while a red color indicates a high frequency of the pattern. Other ranges of blue and yellow indicate intermediate frequencies. Each row in the matrix represents an image and each column represents a visual pattern.

We use agglomerative hierarchical clustering, with average linkage, to organize both, rows and columns in the matrix and the corresponding dendrogram is also drawn alongside the matrix representation. The distance measure applied in this work is Euclidean distance among bag-of-features representations (rows) and the occurrence of visual patterns in all images (columns). This analysis is expected to organize rows such that images in each group share a semantic concept. It highly depends on the bag-of-features representation, so that we can evaluate how good this representation is for semantic image contents. In addition, the column organization is expected to reveal the set of visual patterns that are related to particular semantic concepts.

4 Results

The image dataset used in this work is a set of histopathology images used to diagnose a special skin cancer known as basal cell carcinoma. This dataset has been used in previous studies for automatic image annotation and retrieval [1]. A subset of this collection has been selected to analyse the structure of 4 histopathology concepts (cystic change, lesion with fibrosis, morpheaform pattern and pilosebaceous annexa). This subset of images sums up to 348 images processed for this study (67, 90, 37 and 154 for each concept class respectively).

4.1 Correlation Analysis

The correlation analysis shows that some visual words are more relevant to identify some particular concepts than others. Figure 2 shows how the four concepts

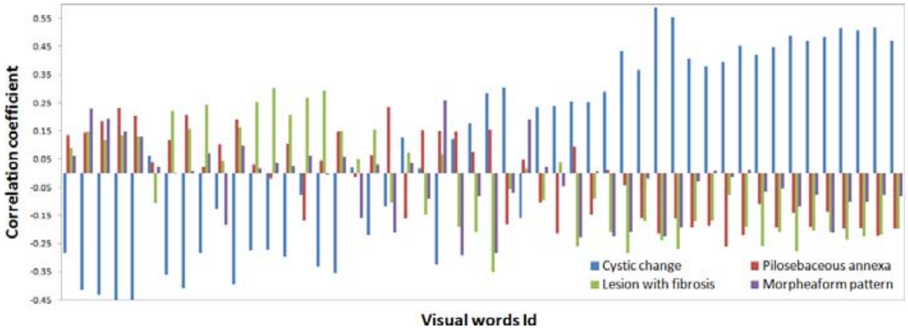


Fig. 2. Correlation coefficient measures between high-level concepts and visual words. Visual words in horizontal axis are sorted by frequency of occurrence from left to right in descending order.

Table 1. Ten visual words with highest correlation value for each concept

Concept	1	2	3	4	5	6	7	8	9	10
Cystic change										
Pilosebaceous annexa										
Lesion with fibrosis										
Morpheaform pattern										

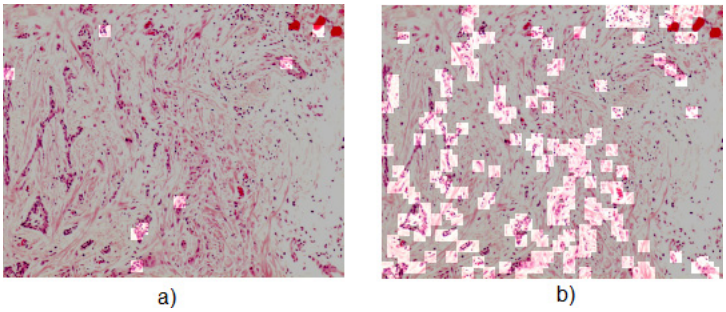


Fig. 3. Spatial location of visual patterns in an image in the category *lesion with fibrosis*. a) highlighted blocks are the ten most correlated visual patterns for the *cystic change* concept. b) highlighted blocks are the ten most correlated visual patterns for *lesion with fibrosis*.

are correlated with each visual pattern. Note that *cystic change* is highly correlated with a set of visual patterns that other concepts are not. It can also be observed for *lesion with fibrosis.* For all concepts it is possible to identify a set of highly correlated visual patterns, since the plot in general shows that patterns with high correlation with a concept present low correlation with others.

Table 1 shows top the ten visual words with highest correlation for each concept. The correlation analysis assigns to each concept a set of visual words. *Cystic change*, for example, is more correlated with dark elements and parts of big circular patterns, which is consistent with a notion of large and dense cells and nuclei. On the other hand, *Lesion with fibrosis* shows small gray points over a bright background.

Figure 3 shows the spatial location of visual patterns on an image of the category *lesion with fibrosis.* Relevant visual word are shown as blocks with a lighter color. Subfigure 3.a) highlights the top-ten visual patterns from the *cystic change* category, showing a low presence of those patterns. On the other hand Subfigure 3.b) highlights the top-ten visual patterns of the *lesion with fibrosis* category, which are clearly more frequent in the image.

4.2 Cluster Analysis

Cluster analysis allows to distinguish groups of related visual patterns and a general organization of images and concepts in the collection under the bag-of-features representation. It is achieved using a graphical representation of the data, indicating occurrence values in a colored matrix. Colors range from dark blue, indicating a very low frequency, to red, indicating high frequency values. To plot this matrix, the 6 most frequent visual words were ignored since they usually correspond to background and do not have discriminative power. Figure 4 shows the obtained matrix for all images in the analysed collection, with visual patterns from the codebook organized in columns, and images organized in rows. The clustering algorithm reordered rows and columns according to their similarity.

This matrix shows group of images related to groups of visual patterns. For instance, in Figure 4 a red box and a black box in the upper-left corner of the matrix shows two different groups of images with a high frequency of several visual patterns. In the vertical dendrogram these groups are colored with green and blue respectively and all of the images in them present the *cystic change* concept. The left side of the figure shows the images and the visual words associated with those regions of the cluster matrix. The orange box, in the same figure, shows how other images in the red portion of the vertical cluster present a high frequency for other visual words. In this group, there are images with other concepts, mainly *pilocebaceus annexa* and l*esion with fibrosis.*

The cluster analysis shows that it is possible to find visual patterns that can be associated with semantic concepts. The visual representation makes it easier the task of finding those visual patterns. In this particular example, the class of images tagged with the *cystic change* concept are clearly differentiated from the other classes by a characteristic set of low-level visual patterns associated with large cells and nuclei.

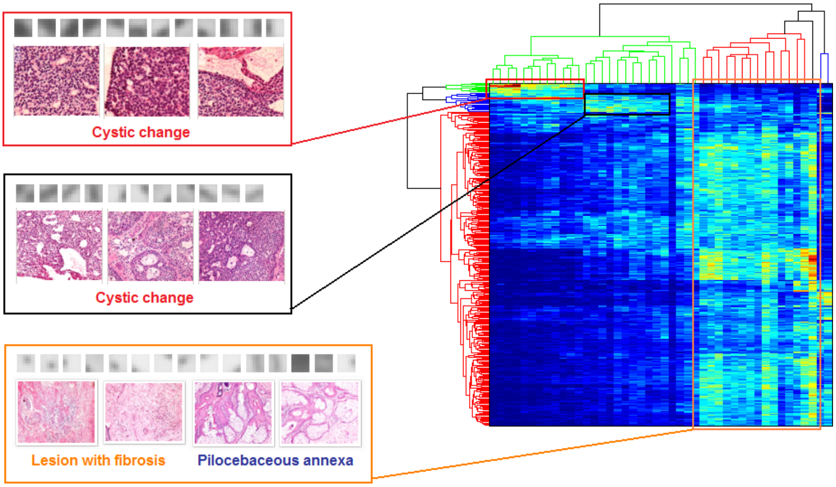


Fig. 4. Cluster analysis on the complete image dataset with 4 concepts. The rows of the matrix correspond to images and the columns correspond to visual words. The color of the matrix represent the frequency of the visual words for each image: blue represents low frequency, red represents high frequency. Both, images and visual words are clustered using hierarchical clustering. The result is represented by the vertical and horizontal dendograms. Three different regions of the matrix are marked by colored boxes. The corresponding visual words, concepts and sample images of these regions are detailed in the left side rectangles.

5 Conclusions and Future Work

This paper has presented a framework to identify and analyse visual patterns in a collection of medical images using a bag-of-features representation. The main hypothesis of this paper was that visual words, identified in the collection using the bag-of-features representation, can be related to semantic concepts in histopathology images. The hypothesis was corroborated by the exploratory experiments based on correlation and cluster analysis. These results suggest that this representation may be useful for analysis and understanding of histopathology images. The cluster analysis is analogous to the one used in bioinformatics to analyse gene array data, where the goal is, e.g., to find how a diseases relates to the presence or absence of a particular gene. In the image analysis context, visual words are analogous to genes with the important advantage that they could be directly related to specific regions of particular images. This kind of analysis is not possible with other image descriptors such as moments, histograms or transformation coefficients. In addition, these analysis may help to design and improve automatic tools to manage image collections, such as image retrieval systems. For instance, understanding the group of visual patterns that better describe a set of concepts, weighting schemes or pruning strategies may be applied in a more informed fashion.

References

1. Caicedo, J.C., Cruz, A., Gonzalez, F.: Histopathology image classification using bag of features and kernel functions. In: Combi, C., Shahai, Y., Abu-Hanna, A. (eds.) Artificial Intelligence in Medicine (AIME 2009). LNCS (LNAI), vol. 5651, pp. 126–135. Springer, Heidelberg (2009)
2. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision (2004)
3. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 95(25), 14863–14868 (1998)
4. Iakovidis, D., Pelekis, N., Kotsifakos, E., Kopanakis, I., Karanikas, H., Theodoridis, Y.: A pattern similarity scheme for medical image retrieval. IEEE Transactions on Information Technology in Biomedicine (2008)
5. Ogiela, M.R., Tadeusiewicz, R.: Artificial intelligence structural imaging techniques in visual pattern analysis and medical data understanding. Pattern Recognition 36(10), 2441–2452 (2003)
6. Orabona, F., Caputo, B., Tommasi, T.: Clef 2007 image annotation task: An svm - based cue integration approach. In: Working Notes of the 2007 CLEF Workshop, Budapest, Hungary (2007)
7. Yang, G., Yu, X., Zhuang, X.: The current status and development of pattern recognition diagnostic methods based on medical imaging. In: IEEE International Conference on Networking, Sensing and Control, 2008. ICNSC 2008, pp. 567–572 (2008)

A Brief Index for Proximity Searching

Eric Sadit Téllez[1], Edgar Chávez[1,2], and Antonio Camarena-Ibarrola[1]

[1] Universidad Michoacana
[2] CICESE

Abstract. Many pattern recognition tasks can be modeled as proximity searching. Here the common task is to quickly find all the elements *close* to a given query without sequentially scanning a very large database.

A recent shift in the searching paradigm has been established by using *permutations* instead of distances to predict proximity. Every object in the database record how the set of reference objects (the permutants) is *seen*, i.e. only the relative positions are used. When a query arrives the relative displacements in the permutants between the query and a particular object is measured. This approach turned out to be the most efficient and scalable, at the expense of loosing recall in the answers. The permutation of every object is represented with κ short integers in practice, producing bulky indexes of $16\kappa n$ bits.

In this paper we show how to represent the permutation as a binary vector, using just one bit for each permutant (instead of $\log \kappa$ in the plain representation). The Hamming distance in the binary signature is used then to predict proximity between objects in the database. We tested this approach with many real life metric databases obtaining faster queries with a recall close to the Spearman ρ using 16 times less space.

1 Introduction

A metric space is composed by an universe of objects $\mathbb{U}$, and a distance function $d : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$, such that for any $x, y, z \in \mathbb{U}$, $d(x, y) > 0$, $d(x, y) = 0 \iff x = y$, $d(x, y) = d(y, x)$ (symmetry), and obeying the triangle inequality: $d(x, z) + d(z, y) \geq d(x, y)$.

Some common tasks require distances expensive to compute (i.e. comparing fingerprints, searching by content in multimedia, etc) and hence sequential scan does not scale for large problems. Proximity queries are usually of two types, for a given database $S \subseteq \mathbb{U}$ with size $|S| = n$, $(q \in \mathbb{U}, r \in \mathbb{R})_d = \{x \in S \mid d(q, x) \leq r\}$, denote a *range query*. The other type of query is the focus of this paper, the k *nearest neighbor*, denoted $kNN_d(q)$, which retrieve the k closest elements to q in S, formally it retrieves the set $R \subseteq S$ such that $|R| = k$ and $\forall u \in R, v \in S - R$ it follows $d(q, u) \leq d(q, v)$.

Most indexes use the triangle inequality to avoid a sequential scan. Upper bounds of the distance between the query and the database objects can be obtained by computing some distances beforehand to the so-called *pivots* or by dividing the space in regions with the so-called *compact partitioning* indexes. Due to space restrictions we do not overview current approaches, nevertheless a

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 529–536, 2009.

deeper and extended catalog for searching in metric spaces can be found in [1,2,3]. We will focus on the permutations index (described in detail below) because it has shown to be very scalable, indexing hundreds of millions of images in the Cophir project [6].

1.1 Overview of the Permutations Based Index

The motivation behind this indexing method [4] is to shift the problem of comparing directly the query object against every object in the database to comparing the *perspective* in which a set of elements are perceived. Each database element has an unique perspective of the *permutants* (defined below) and the query is only compared to those elements having similar perspective of the permutants.

Let $\mathbb{S}$ be the database of objects, and $\mathbb{P} \subseteq \mathbb{S}$ be a set of distinguished objects from the database, called *permutants*. Assume x is the query. Each x defines a *permutation* Π_x, where the elements of $\mathbb{P}$ are written in increasing order of distance to x. Ties are broken using any consistent order, for example, the order of the elements in $\mathbb{P}$.

Definition 1. *Let* $\mathbb{P} = \{p_1,\ p_2, \ldots, p_k\}$ *and* $x \in \mathbb{X}$*. Then we define* Π_x *as a permutation of* $(1 \ldots k)$ *so that, for all* $1 \leq i < k$ *it holds either* $d(p_{\Pi_x(i)}, x) < d(p_{\Pi_x(i+1)}, x)$*, or* $d(p_{\Pi_x(i)}, x) = d(p_{\Pi_x(i+1)}, x)$ *and* $\Pi_x(i) < \Pi_x(i+1)$*.*

Each database element u will be represented by a permutation Π_u. The query will be represented by Π_q using the same definition. Elements that are close will have similar permutations. Defining a similarity is central to obtain good results. An excellent predictor is the Spearman Rho, defined as the sum the squares of differences in the relative positions of each element in both permutations. That is, for each $p_i \in \mathbb{P}$ we compute its position in Π_u and Π_q, namely $\Pi_u^{-1}(i)$ and $\Pi_q^{-1}(i)$, and sum up the squares of the differences in the positions [4]. Formally defined below in 2.

Definition 2. *Given permutations* Π_u *and* Π_q *of* $(1 \ldots k)$*, Spearman Rho is defined as*

$$S_\rho(\Pi_u, \Pi_q) \;=\; \sum_{1 \leq i \leq k} \left(\Pi_u^{-1}(i) - \Pi_q^{-1}(i)\right)^2 .$$

We use the same example depicted in [4] for illustrating the definition of $S_\rho(\Pi_q, \Pi_u)$. Let $\Pi_q = 6, 2, 3, 1, 4, 5$ be the permutation of the query, and $\Pi_u = 3, 6, 2, 1, 5, 4$ that of an element u. A particular element p_3 in permutation Π_u is found two positions off with respect to its position in Π_q. The differences between permutations are: $1-2, 2-3, 3-1, 4-4, 5-6, 6-5$, and the sum of their squares is $S_\rho(\Pi_q, \Pi_u) = 8$.

Note that we can compute $S_\rho(\Pi_q, \Pi_u)$ by obtaining the inverse of both permutations and then computing the Euclidean distance of the inverse. It is also shown in [4] that we can use the sum of the absolute of the differences, without the squares, without noticeable penalization in the index recall.

The result is a table of n rows (one per database element) and k columns (one per permutant). Each cell needs $\lceil \log_2 k \rceil$ bits to store one permutation at each row. The indexing cost is kn distance computations plus $O(nk \log k)$ CPU time to sort all the permutations.

The search has two phases. The first sorts the database according to the permutation distance and selects as candidates the first elements. The second phase is to check the list. The permutation index allows kNN searches in pseudo-metric spaces, because the triangle inequality is not used explicitly. Our technique inherits this property allowing faster searches and smaller indexes.

2 The Brief Permutations

Our goal is to achieve the same performance of the permutations based index using only one bit to represent each permutant. The Algorithm 1 shows the algorithm *Encode* for condensing the permutation information into bit strings. In *Encode* we can note that for big enough m (e.g $m \geq \frac{|P|}{2}$) the permutants in the center will be rarely set to 1. In order to reduce this effect we compute a second swapped permutation codifying them in the same bit string, as depicted in Algorithm 2.

Algorithm 1. Bit-encoding of the permutation P under the module m

```
Encode(Permutation $P$, Positive Integer $m$)
Let $P^{-1}$ be the inverse $P$.
$C \leftarrow 0$ {Bit string of size $|P|$, initialized to zeros}
for all $i$ from $0$ to $|P| - 1$ do
    if $|i - P^{-1}[i]| > m$ then
        $C[i] \leftarrow 1$
    end if
end for
return C
```

The brief index encodes all the objects in the database in different bit-strings, the Hamming distance is used to compare objects, instead of the Spearman Rho. The searching is shown in Algorithm 3, I is the brief index. Computing Hamming distances is way faster than computing the Spearman Rho, and this is the only operation needed to satisfy queries.

To fix ideas, consider the following example. Let $m = 2$, $u = (3,6,2,1,5,4)$, $r = (5,3,1,6,2,4)$ and $q = (6,2,3,1,4,5)$. After the inverse $u^{-1} = (4,3,1,6,5,2)$, $r^{-1} = (3,5,2,6,1,4)$ and $q^{-1} = (4,2,3,5,6,1)$. Applying algorithm 1 we have $\hat{u} = (|1-4| > m, |2-3| > m, |3-1| > m, |4-6| > m, |5-5| > m, |6-2| > m) = (1,0,0,0,0,1)$, supposing $|a - b| > m$ evaluates to 1 for true and 0 for false. Similarly, we obtain $\hat{r} = (0,1,0,0,1,0)$ and $\hat{q} = (1,0,0,0,0,1)$. If H is the hamming distance, $H(\hat{u},\hat{q}) = 0$ and $H(\hat{r},\hat{q}) = 2$. Clearly, q is the closer one to u, and this can be verified using S_ρ as $S_\rho(u,q) = 8$, and $S_\rho(r,q) = 46$.

Algorithm 2. Bit-encoding using permutation of the center. Interchangeable with Encode.

```
EncodePermCenter(Permutation $P$, Positive Integer $m$)
Let $P^{-1}$ be the inverse $P$
$C \leftarrow 0^{|P|}$ {Bit string of size $|P|$, initialized to zeros}
$M = \frac{|P|}{4}$
for all $i$ from 0 to $|P|-1$ do
    $I \leftarrow i$
    if $\lfloor \frac{I}{M} \rfloor \bmod 3 \neq 0$ then
        $I \leftarrow I + M$
    end if
    if $|I - P^{-1}[i]| > m$ then
        $C \leftarrow C|(1 << i)$
    end if
end for
return $C$
```

The binary mapping works because in essence it reproduces the same behavior than S_ρ with coarse granularity using two possible values (i.e. 0 and 1). Suppose two vectors u and v and we want to select the closest vector to q. In the brief representation we are neglecting (not adding) the displacements smaller than a certain threshold. This is compensated by using a larger number of permutants in the index.

As we are using bit encoded values we can use XOR bit-operation ($\oplus$) using the bit parallelism inherent in the computer integer operations computing 32 or 64 operations per instruction instead of the most expensive operations difference and product used in the S_ρ. The count of the enabled bits can be calculated using a previously calculated table for one or two bytes.

We can resume that $0 \oplus 0 = 0$ means an small movement difference, $0 \oplus 1 = 1$ meaning a big difference. $1 \oplus 1$ can significant a really big difference or an small one, in order to encode in just one bit each permutant we choose only one, and to be able to use the hamming distance we choose $1 \oplus 1 = 0$. Choosing $1 \oplus 1 = 1$ can be efficiently computed using $\oplus$ as OR instead of XOR.

Algorithm 3. Procedure to search kNN for q

```
SearchKNN(Hamming Index $I$, Permutants $\mathbb{P}$, Distance $d$, Object $q$, Positive Integer $m$, Positive Integer $k$, Positive Integer $Cand$)
$P \leftarrow$ Get permutation for $q$ under $\mathbb{P}$ and $d$.
$\hat{h} \leftarrow Encode(P, m)$.
$R \leftarrow$ Retrieve the $lNN$ for $\hat{h}$ with metric index $I$ using $l = Cand$ {Remember that $R \subseteq S$}.
$Res \leftarrow [\,]$
for all $s \in R$ do
    $Res \leftarrow Res + [(d(s,q), s)]$
end for
$Res \leftarrow$ sort $Res$ by the first argument in the tuple, keep the smallest $k$ results.
return Res
```

3 Experiments

The tested databases were taken from the *metric space* library[1] and the natix project's site[2]. Our implementation is available as open source from

[1] Metric space library www.sisap.org

[2] Natix web site is www.natix.org

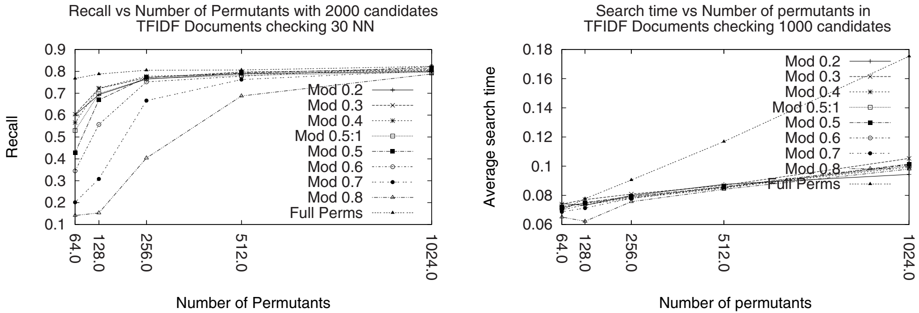
www.natix.org. All indexes share the same distance function's implementation. The experiments were performed in a laptop computer with Intel Core 2 at 2.4 GHz and 2GiB of RAM, running MacOS X 10.5.6. The indexes run in main memory and without parallelization. We have tested our index in a large number of real life databases, due to space constrains we show the results in three databases; documents for textual information retrieval, color's histogram vectors for multimedia information retrieval, and fingerprints of songs used for music information retrieval. Please note that the brief representation of the permutant space allows keeping the index in main memory. Secondary memory access is only needed when checking candidates.

The presented results compares the full permutation index and a set of brief indexes using different modules, the modules are presented in the Figures as the ratio between $\frac{m}{|P|}$. The Encode algorithm is used for all of them, except for curves named *Mod 0.5:1* whose are based on the EncodePermCenter algorithm.

3.1 Documents

A collection of 25157 short news articles in the $TF \times IDF$ format from Wall Street Journal 1987 – 1989 files from TREC-3 collection. We use the angle between vectors as distance measure [5].

We extracted 100 random documents as queries, these documents were not indexed. Each query searches for 30 nearest neighbors (a metric index, like BKT [2] needs to check up to 98% of the database for this task). Figure 1(a) shows the recall for 30NN. Please note that the number of distance computations is the number of permutants plus the number of candidates, then for $30NN$ recall of 0.82 we need to review only the 6% of the database instead of the 98% in the alternative metric index (not shown for space constrains). If instead of $30NN$ we search for the nearest neighbot the recall increases to 97%. The recall for the brief permutations is closely related with the module, also note that as the number of permutants increases the module effect decreases. We can see that module 0.5



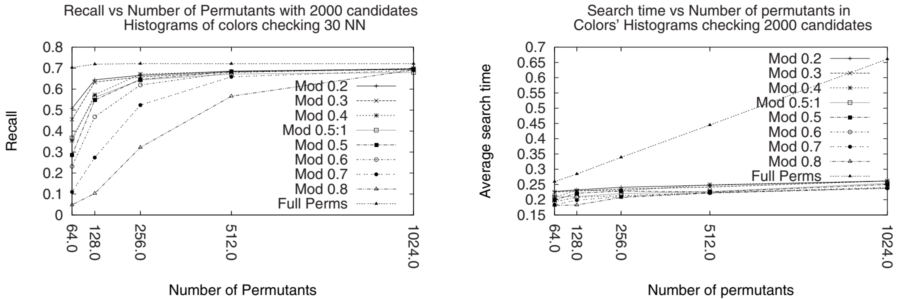
(a) Recall for 30NN (b) Average search time for documents

Fig. 1. Experiment results for brief index against full permutations using the documents $TF \times IDF$ collection and vector's angle as distance

is a fair choice for any number of permutants. We can see in both recall Figures that the brief index performs slightly better than the full permutants. Figure 1(b) shows the average time per search needed for each number of permutants, naturally the brief representation is faster.

3.2 Vectors

We selected a set of 112544 color histograms (112-dimensional vectors) from an image database[3] We choose randomly 200 histogram vectors and we applied a *perturbation* of ±0.5 on one random coordinate. The search consist on finding 30NN under L2 distance. The BKT needed to check 65% of the database. We achieved only a recall of 0.7 for 2000 checked candidates (equivalent to review a 2% of the database). This behavior is inherent to the permutations based index, the exact reasons of this behavior is unknown, but this experiment shows that the behavior is inherited by the brief index. Even with this *poor* recall, it's an excelent approximation for achieving fast searches for massive Multimedia Information Retrieval approaches [6] where a recall of 0.5 is reported for a larger database.



(a) Recall for 30NN

(b) Average search time for color histograms

Fig. 2. Results for brief index against full permutations using the color's histogram collection and L2 distance

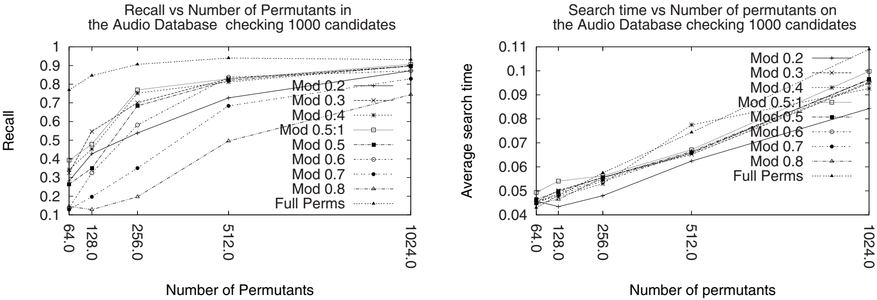
The behavior of this example is similar to the previous experiment, better times, less space for the same task.

3.3 Audio Fingerprints

A database of 10254 *multi-band spectral entropy signature* (MBSES) [7] using three byte's frame for each 46 ms. The signatures were extracted from full songs of assorted genre[4]. We use a non-metric distance called *probabilistic pairing psudo*

[3] The original database source is `http://www.dbs.informatik.uni-muenchen.de/~seidl/DATA/histo112.112682.gz`

[4] The fingerprint's database is available from the `www.natix.org` website.



(a) Recall for matching the NN against the ground truth

(b) Average search

Fig. 3. Results for brief index against full permutations for the audio collection

metric [8] which is defined as the minimum hamming distance from one short sequence of length m against all m-grams inside a larger sequence. The distance's cost is $O(m \times (n - m + 1))$. We use excerpts of 20s as permutants and degraded excerpts of 20s as queries, both sets are disjoint.

Figure 3(a) shows a recall of 0.92 for the full permutation index, and 0.83 for the brief index using 512 permutants. Please note that BKT can be used at the expense of loosing some results because the probabilistic pairing pseudo metric do not follow the triangle inequality. The BKT gives a recall above 0.9 reviewing more than 40% of the database, resulting in 30 seconds per search. The brief index needs to review 512 distance's evaluations to compute the permutation, and 1000 distances verification (i.e. review 10% of the database, note that this is possible because permutants and queries have the same length). The verification is done using the transitivity kept by the distance, using only 12 or 24 frames, reducing the final cost of the query, Figure 3(b) shows the time per search.

4 Conclusions and Future Work

We have presented a new indexing method based on permutations. Our representation is able to use only one bit for each permutant, opposed to the 16 bit usual representation without noticeable impact in the recall of the index and 4 to 12 faster than full permutations. We are working on experiments in very large databases with an specialized indexing for Hamming distance for speeding up the searches (in the paper the times shows the effect in sequential scanning in the brief permutation space). Although *module* 0.5 is a good choice for any space and any number of permutants (specially when swapping center permutants as described in algorithm 2), a more careful tuning of the *modulus* used for obtaining the brief representation is needed.

Acknowledgments

We wish to thank the comments from the referees and from Rodrigo Paredes from University of Talca, in Chile, helping to improve the presentation.

References

1. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers, San Francisco (2006)
2. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L.: Searching in metric spaces. ACM Comput. Surv. 33(3), 273–321 (2001)
3. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys 33(3), 322–373 (2001)
4. Chavez, E., Figueroa, K., Navarro, G.: Effective proximity retrieval by ordering permutations. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(9), 1647–1658 (2008)
5. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
6. Amato, G., Savino, P.: Approximate similarity search in metric spaces using inverted files. In: InfoScale 2008: Proceedings of the 3rd international conference on Scalable information systems, ICST, Brussels, Belgium, Belgium, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 1–10 (2008)
7. Ibarrola, A.C., Chávez, E.: A robust entropy-based audio-fingerprint. IEEE, Los Alamitos (2006)
8. Chavez, E., Camarena-Ibarrola, A., Téllez, E.S., Bainbridge, D.: A permutations based index for fast and robust music identification. Technical Report. Universidad Michoacana (2009)

Pigmented Skin Lesions Classification Using Dermatoscopic Images

Germán Capdehourat[1], Andrés Corez[1], Anabella Bazzano[2], and Pablo Musé[1]

[1] Departamento de Procesamiento de Señales, Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, Uruguay
[2] Unidad de Lesiones Pigmentadas, Cátedra de Dermatología, Hospital de Clínicas, Facultad de Medicina, Universidad de la República, Uruguay

Abstract. In this paper we propose a machine learning approach to classify melanocytic lesions in malignant and benign from dermatoscopic images. The image database is composed of 433 benign lesions and 80 malignant melanoma. After an image pre-processing stage that includes hair removal filtering, each image is automatically segmented using well known image segmentation algorithms. Then, each lesion is characterized by a feature vector that contains shape, color and texture information, as well as local and global parameters that try to reflect structures used in medical diagnosis. The learning and classification stage is performed using AdaBoost.M1 with C4.5 decision trees. For the automatically segmented database, classification delivered a false positive rate of 8.75% for a sensitivity of 95%. The same classification procedure applied to manually segmented images by an experienced dermatologist yielded a false positive rate of 4.62% for a sensitivity of 95%.

1 Introduction

The incidence of melanoma in the general population is increasing worldwide. It is estimated that by the end of this decade, four million new melanomas will be diagnosed in the world, causing the death of half million people. If early diagnosed and treated, the mean life expectancy of these individuals would have been enlarged by at least 25 years. Because advanced cutaneous melanoma is still incurable, early detection, by means of accurate screening, is an important step toward mortality reduction. Detection of thin malignant melanoma is the most effective way to avoid mortality related to this disease.

Dermoscopy is a noninvasive in vivo technique that assists the clinician in melanoma detection in its early stage. Images are acquired using epiluminescence light microscopy, that magnifies lesions and enables examination down to the dermo-epidermal junction. This permits to visualize new morphologic features and in most cases facilitates early diagnosis. However, evaluation of the many morphologic characteristics is often extremely complex and subjective [1].

Advances in objective dermatology diagnosis were obtained in 1994 with the introduction of the ABCD rule [2]. The ABCD rule specifies a list of visual features associated to malignant lesions (Asymmetry, Border irregularity, Color

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 537–544, 2009.

irregularity and presence of Dermoscopic structures), from which a score is computed. This methodology provided clinicians with a useful quantitative criterion, but it did not prove efficient enough for clinically doubtful lesions (CDL). The main reason for this is the difficulty in visually characterizing the lesions' features. Setting an adequate decision threshold for the score is also a difficult problem; by now it has been fixed based in several years of clinical experience. Many authors claim that these thresholds may lead to high rates of false diagnoses [3]. An alternative algorithm for melanocytic lesion diagnosis is the 7-points checklist [4]. This algorithm consists of analyzing the presence of the seven most important color or geometric stuctures that characterize malignant melanoma (blue whitish veil, atypical pigment network, irregular streaks, etc.).

The computerized analysis of dermatoscopic images can be an extremely useful tool to measure and detect sets of features from which dermatologists make their diagnosis. It can also be helpful for primary screening campaigns, increasing the possibility of early diagnosis of melanoma. Currently there is no commercial software for massive use in clinical practice. Our ultimate goal is to develop software for the recognition of early-stage melanomas, based on images obtained by digital dermoscopy. This would enable unsupervised classification of melanocytic lesions, assigning a confidence index for each classification. The result of such classification procedure will separate the "screened" lesions in two groups. The first group corresponds to lesions that were classified with high enough confidence level, while the second one corresponds to those lesions for which the confidence level is low and consequently, requires subsequent inspection by an experienced dermatologist. In this sense, the classification technique is actually a semi-automated method.

The paper is organized as follows. In Section 2 we present a brief overview of previous related work. In Section 3 we describe the composition of our database of dermatoscopic images, and in Section 4 we present our approach to melanocytic lesions classification. Results and performance are presented and discussed in Section 5. We conclude in Section 6.

2 Computerized Analysis of Dermoscopic Images: State of the Art

Computer aided image analysis in skin lesion diagnosis is a relatively new research field. While the first related work in the medical literature seems to date back to 1987 [5], its contribution was limited since by that time computer vision and machine learning were both emerging fields (the first edge detectors where starting to appear). One of the first significant contributions from the image processing community was reported in [6]. In this work, the authors propose a classical machine learning approach for dermatoscopic image classification. The first stage is automatic color-based lesion segmentation. Then, over a hundred features are extracted from the image (shape and color, and gradient distribution in the neighbourhood of the lesion boundary). Feature selection was obtained using sequential forward and sequential backward floating selection.

Classification experiments, performed with a 24-NN classifier, delivered a sensitivity of 77% with a specificity of 84%.

To our knowledge, up to now the best results in automated melanocytic lesion classification where obtained by Celebi *et al.* [7]. See this reference for a complete summary of the results obtained by key studies from 2001 onwards, along with their database sizes. As in [6], the proposed approach is a classic machine learning methodology. After an Otsu-based image segmentation, a set global features are computed (area, aspect ratio, asymmetry and compactness). Local color and texture features are computed after dividing the lesion in three regions: inner region, inner border (an inner band delimited by the lesion boundary) and outer border (an outer band delimited by the lesion boundary). Feature selection is performed using ReliefF [8] and CFS algorithms [9]. Finally, the feature vectors are classified into malignant and benign using SVM with model selection [10]. Performance evaluation gave a specificity of 92.34% and a sensitivity of 93.33%.

3 Database Composition

Our database is composed of 513 images of melanocytic lesions: 433 benign lesions and 80 malignant melanoma. Among the set of benign lesions, over a hundred correspond to dysplastic melanocytic nevi. It is important to note that in general these kind of lesions are the benign lesions that are visually the most alike to malignant melanoma; many of them are clinically doubtful for experienced dermatologists. This composition was based on the existence of dermatoscopic and histopathologic studies, which were used as ground truth for the classification procedure. Actually, the original database was larger, but some images were discarded for the following reasons: the images do not capture the whole lesion, poor image quality or excessive presence of hair.

Every image in this database has been manually segmented by a dermatologist, who also provided dermatoscopic diagnosis based on the ABCD rule and the 7-points checklist. This enables performance evaluation for both segmentation and features' measurements.

4 Dermoscopic Images Classification: Proposed Approach

Our approach follows a typical machine learning methodology. In the first stage, we tackle image processing problems such as image filtering, restoration and automatic segmentation to isolate the lesion's area. The second stage consists of extracting features from the image for further lesion classification into malignant or benign. Features are inspired by the same elements that dermatologists use for lesion diagnosis. Once lesions' features have been extracted, labeled lesions are used to train a meta-classifier obtained using boosting based on decision trees. Classification errors and ROC curves are obtained by means of cross validation. In this section we give details of each of these stages.

4.1 Preprocessing and Hair Removal

Lesion segmentation in the presence of hair is usually doomed to failure. Thus, previous application of a hair removal filter is unavoidable. Automatic hair removal requires hair detection and image inpainting. We used Dullrazor [11], a well known algorithm for hair removal. This algorithm identifies the image segments that approximate the structure of the hair, and then the regions that contain these segments are interpolated using the information of the surrounding pixels. A typical result is shown in Figure 1(a)(b). For the inpainting part, more sophisticated techniques were also explored, with similar results.

4.2 Segmentation

Segmentation of melanocytic lesions can be an extremely hard problem. Besides the presence of hair, many lesions present diffuse borders, that can be difficult to determine even for dermatologists. Several methods of image segmentation were explored, based on edge detection and on region information. In general it is appropriate to combine different features (texture, edges, color) for better results. Methods combining these sources of information were also studied. Among the variational methods family, we considered Otsu using color norm instead of grey level [12], Mumford-Shah [13], Geodesic Active Contours and Geodesic Active Regions [14]. We explored also several methods based on the topographic map, using both boundary and color and texture region information [15,16]. We are currently investigating spectral clustering – graph based approaches.

Overall, none of the methods outperformed the others. We decided to use the color-based Otsu method for it is simpler and significantly faster. Of course, there are pathological cases in which it fails, and sometimes one of the others provides satisfactory results. This suggests that a software for clinical use should propose the choice of a few candidate segmentations to the user in case they differ.

4.3 Feature Extraction

A set of global measurements of shape (aspect ratio, symmetry, compacity, etc.) and border irregularity were computed from each lesion. More localized features of texture and color distribution were also extracted. Previous to their extraction, each lesion is decomposed into three sub-regions: the interior and the outer and inner border (Figure 1). For each of these regions, the color features consist of some statistics of its distribution (mean and variances per channel in RGB and HSV spaces), and the texture features based on Gabor filters capture information of local contrast, correlation, heterogeneity and energy. For each lesion, a total number of 57 features are extracted.

Note that information concerning the presence or absence of several geometric patterns that are relevant to the 7 points checklist is not included in the feature vectors. This requires the detection of these structures, which is not a trivial task, what explains why they are not included in any previous work, either. We are currently investigating these detection problems, for we are confident that the capability of detecting this structures will boost our method performance.

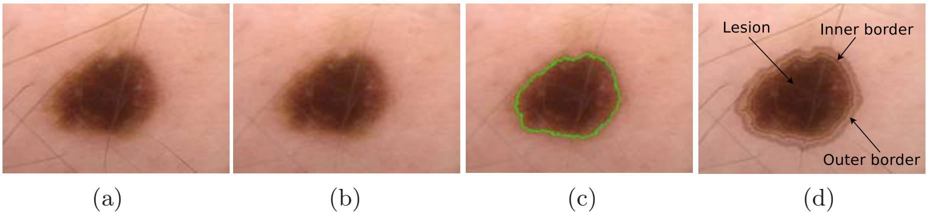


Fig. 1. (a) Original lesion. (b) Result of the hair removal filter. (c) Color-based Otsu segmentation. (d) Definition of the three regions used for feature extraction.

4.4 Classification

The goal of this stage is to classify the feature vectors in two classes: malignant and benign. A classification technique that prove very successful in our experiments consist of performing decision trees combination *via* adaptive boosting. Boosting exploits the inherent instability in learning algorithms by combining multiple models, in a way that models complement one another. This is achieved by assigning weights to the training data, and modifying them after each classifier by increasing the weight of misclassified samples, and decreasing these of correctly classified ones. Hence, after each iteration, a new classifier is forced to focus on classifying the hard samples correctly. The algorithm finishes after a user-defined number of T iterations, that generates a set of T classifiers. To each of them, a weight that increases with its performance is associated. Classification of new unlabeled data is performed by a weighted vote of the T classifiers.

The algorithms we considered for the classification framework are C4.5 decision trees [17], and AdaBoost.M1 [18], using Weka's implementations. In order to deal with class imbalance, we applied a widely used synthetic over-sampling technique (SMOTE [19]) to the minority class.

5 Results

Performance evaluation was conducted using 10 times - 10 fold cross-validation. To assess the impact of the learning and classification method, we compared our results with SVM with model selection (preceeded by ReliefF feature selection). As in [7], a RBF kernel was used, and optimal parameters (the weight that controls model complexity and the RBF parameter) were obtained by grid search optimization with 10 fold cross-validation. Classification performance was also estimated using 10 times - 10 fold cross-validation.

The same experiments were repeated, replacing automatic segmentation by manual segmentation by a dermatologist. This was carried on to assess the influence of automatic segmentation errors.

The left plot in Figure 2 shows the overall system performance using automatic segmentation, for both learning strategies. The right plot shows the results for the manually segmented images. In both cases, the AdaBoost/C4.5 method

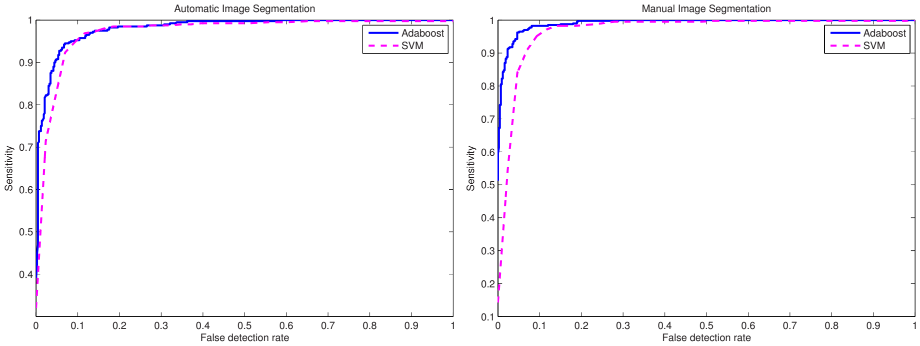


Fig. 2. Left: ROC curves for the AdaBoost/C4.5 and SVM approaches for automatically segmented (left) and manually segmented (right) images. See text for details.

Table 1. Performance indicators for the ROC curves in Figure 2

Method	FPR for 95% sensitivity	Area under ROC
Automatic segmentation, AdaBoost - C4.5	8.75 %	0.981
Automatic segmentation, SVM	9.52 %	0.963
Manual segmentation, AdaBoost - C4.5	4.62%	0.991
Manual segmentation, SVM	9.23 %	0.966

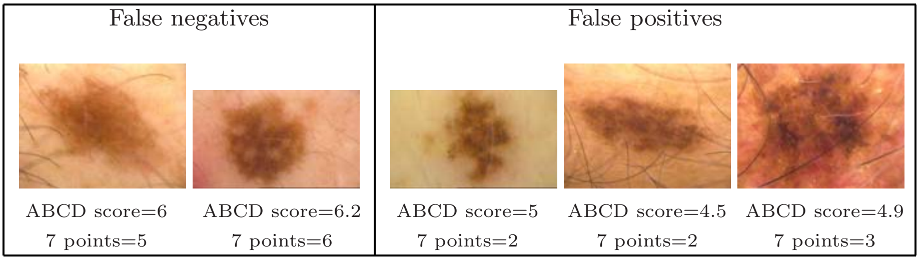


Fig. 3. All misclassified patterns corresponding to lesion images. Color-based Otsu segmentation was used. See text for details.

outperformed the SVM-based approach. Table 1 shows performance indicators for the four experiments.

While the SVM approach using manually or automatically segmented images yielded essentially the same performance, the performance of Adaboost/C4.5 classification of manually segmented images was significantly higher than for the automatically segmented ones. Note that the results we obtained with SVM are slightly better than those reported by Celebi *et al.* [7] (false positive rate of 14% for 95% sensitivity and AUC of 0.966). Our AdaBoost/C4.5 approach shows even higher performance. Note that since the database used by Celebi *et al.* is very similar to ours in size and composition (476 benign lesions and 88 malignant melanoma), this performance comparison makes sense, but only up to a certain point.

Figure 3 shows the five misclassified patterns that correspond to lesion images in the database, for the AdaBoost/C4.5 classification of automatically segmented lesions. Among these lesions, all false positives were dysplastic melanocytic nevi, actually suspicious lesions according to the ABCD rule (CDL scores range from 4.75 to 5.45). Moreover, note that the rightmost one qualifies as melanoma according to the 7-points checklist algorithm (larger or equal than 3 corresponds to malignant melanoma). Concerning the false negatives, posterior inspection by an expert dermatologist revealed subjective overestimation of their scores, since the lesions corresponded to a patient with clinical history of melanoma.

6 Conclusions and Future Work

In this work we presented a machine learning approach to classify melanocytic lesions from dermatoscopic images. The learning and classification stage is performed using AdaBoost.M1 with C4.5 decision trees. Using automatically segmented images, we obtained a false positive rate of 8.75% for a sensitivity of 95%, and an AUC of 0.981. These results are promising and seem to be superior than those reported in the literature. However, performance evaluation is delicate because all reported results were obtained using different databases. At this point, construction of a large database of dermatoscopic images that could be used as reference testbed appears to be a fundamental issue.

Concerning our algorithm, to further improve its performance, methods to detect a larger number of geometry or texture based structures, similar to those used in the 7 points checklist, should be developed. Because of their strong discriminative power, we are confident that the inclusion of these patterns' information in the features vectors will boost the classification results. This is ongoing research and hopefully will be implemented in future versions. It seems also, from the comparison of the results obtained from manually segmented lesions (FPR of 4.62% for a sensitivity of 95%), that errors in automatic segmentation have an important impact and should be reduced. As we pointed out, this is a hard problem since many melanocytic lesions show highly diffuse contours. Note, however, that nothing prevents us to manually segment the training database, and to propose to the user, for each new lesion, the choice of candidate segmentations.

Another interesting related line of research is the characterization of the discriminative power of the considered features. This can be obtained by means of automatic feature selection strategies like the ones that were mentioned here. A rigorous study of this topic, complemented with the comparison of the weights assigned to visual features in the ABCD and other clinical diagnosis rules, may yield useful recommendations to dermatologist for their medical practice.

References

1. Rubegni, P., Burroni, M., Dell'eva, G., Andreassi, L.: Digital dermoscopy analysis for automated diagnosis of pigmented skin lesion. Clinics in Dermatology 20(3), 309–312 (2002)
2. Nachbar, F., Stolz, W., Merkle, T., Cognetta, A., Vogt, T., Landthaler, M., Bilek, P., Braun-Falco, O., Plewig, G.: The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. Journal of the American Academy of Dermatology 30(4), 551–559 (1994)
3. Lorentzen, H., Weismann, K., Kenet, R., Secher, L., Larsen, F.: Comparison of dermatoscopic abcd rule and risk stratification in the diagnosis of malignant melanoma. Acta Derm Venereol 80(2), 122–126 (2000)
4. Johr, R.H.: Dermoscopy: alternative melanocytic algorithms - the abcd rule of dermatoscopy, menzies scoring method, and 7-point checklist. Clinics in Dermatology 20(3), 240–247 (2002)
5. Cascinelli, N., Ferrario, M., Tonelli, T., Leo, E.: A possible new tool for clinical diagnosis of melanoma: The computer. Journal of the American Academy of Dermatology 16(2), 361–367 (1987)
6. Ganster, H., Pinz, A., Rhrer, R., Wildling, E., Binder, M., Kittler, H.: Automated melanoma recognition. IEEE Transactions on Medical Imaging 20, 233–239 (2001)
7. Celebi, M.E., Kingravi, H.A., Uddin, B., Iyatomi, H., Aslandogan, Y.A., Stoecker, W.V., Moss, R.H.: A methodological approach to the classification of dermoscopy images. Comput. Med. Imaging Graph 31(6), 362–373 (2007)
8. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Mach. Learn. 53(1-2), 23–69 (2003)
9. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: ICML 2000: Proceedings of the 7th International Conference on Machine Learning, San Francisco, CA, USA, pp. 359–366 (2000)
10. Schlkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge (2001)
11. Lee, T., Ng, V., Gallagher, R., Coldman, A.: Dullrazor: A software approach to hair removal from images. Computers in Biology and Medicine 27(11), 533–543 (1997)
12. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62–66 (1979)
13. Koepfler, G., Lopez, C., Morel, J.M.: A multiscale algorithm for image segmentation by variational method. SIAM J. Numer. Anal. 31(1), 282–299 (1994)
14. Paragios, N., Deriche, R.: Geodesic active regions: A new framework to deal with frame partition problems in computer vision. Journal of Visual Communication and Image Representation 13, 249–268 (2002)
15. Cao, F., Musé, P., Sur, F.: Extracting meaningful curves from images. Journal of Mathematical Imaging and Vision 22(2-3), 159–181 (2005)
16. Cardelino, J., Randall, G., Bertalmio, M., Caselles, V.: Region based segmentation using the tree of shapes. In: IEEE International Conference on Image Processing, Proceedings (2006)
17. Quinlan, R.J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann, San Francisco (1993)
18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1), 119–139 (1997)
19. Nitesh, V., Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res (JAIR) 16, 321–357 (2002)

A Multi-class Kernel Alignment Method for Image Collection Summarization

Jorge E. Camargo and Fabio A. González

Bioingenium Research Group
National University of Colombia
{jecamargom,fagonzalezo}@unal.edu.co

Abstract. This paper proposes a method for involving domain knowledge in the construction of summaries of large collections of images. This is accomplished by using a multi-class kernel alignment strategy in order to learn a kernel function that incorporates domain knowledge (class labels). The kernel function is the basis of a clustering algorithm that generates a subset, the summary, of the image collection. The method was tested with a subset of the *Corel* image collection using a summarization quality measure based on information theory. Experimental results show that it is possible to improve the quality of the summary when domain knowledge is involved.

Keywords : Image collection summarization, information visualization, clustering, multi-class kernel alignment.

1 Introduction

Effective and efficient access to large collection of images is an important challenge for information retrieval. The main problem is how to find the right images based on their contents (content-based image retrieval, CBIR) [4]. A promising approach to this problem is based on visualization of the whole image collection using a 2D map metaphor. This strategy tries to exploit the human brain capacity for efficiently recognizing visual patterns, so that an ordered display of many images at the same time may help users to find the right information. The visualization is built in such a way that users can see different images distributed in the screen according to their visual similarity and can intuitively start to explore the image collection. In large collections of images it is not possible to show all images to the user due to the limitations of screen devices. Therefore, it is necessary to provide a mechanism that *summarizes* the image collection. This summary represents an overview of the data set and allows the user to start the navigation process. After building this summary, it must be shown to the user, this problem is called *image collection projection*. It is usually addressed using dimensionality reduction methods for obtaining a low dimensional representation of each image that can be projected in a 2D layout [7].

There are some works that have addressed the construction of image collection summaries. Some of them use clustering methods [10,9], similarity pyramids

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 545–552, 2009.

methods [3], graph methods [2,6], neural networks methods [5], among others. In all the cases, the summarization problem is approached as a non-supervised learning problem. Typically, image clusters are identified in the collection and representative images from each cluster are chosen to compose the summary.

This paper proposes a combined supervised and non-supervised strategy for image collection summarization. The supervised part uses domain knowledge, in the form of a training set of labeled images, to build an image kernel function, which can be seen as an image similarity measure. The kernel function is based on individual kernel functions that measure image similarity according to different visual features. The individual kernels are optimally combined using a multi-class kernel alignment strategy. The combined kernel is then used as input for a k-medoids clustering algorithm and the generated medoids correspond to the image collection summary.

The reminder of this paper is organized as follows: In Section 2, the general summarization framework is presented and briefly discussed; in Section 3, the kernel-based approach for improving the summarization is described; Section 4, shows the experimental evaluation of the strategy. Finally, Section 5 presents the conclusions and future work.

2 Image Collection Summarization Framework

We aim to generate an overview of the image collection that faithfully represents the complete collection. To accomplish this objective, we propose the framework shown in Figure 1. The steps of the process are: selection of an image subset for training; extraction of image features; kernel alignment for involving domain knowledge; construction of a combination function based on the parameters found with kernel alignment; clustering using k-medoids for building an image collection summary; and application of a dimensionality reduction technique for generating a 2D visualization of the summary. On the other hand, features of the remaining images are extracted and the kernel function is calculated using the combination function obtained previously. When a new image arrives to the collection, it can be automatically classified in one of the clusters by calculating its similarity with the medoids using the combination function and the image is classified in the cluster of the most similar medoid.

3 Involving Domain Knowledge (Kernel Alignment)

Kernel functions have been successfully used in a wide range of problems in pattern analysis since they provide a general framework to decouple data representation and learning algorithms. A kernel function implicitly defines a new representation space for the input data in which any geometry or statistical strategy may be used to discover relationships and patterns in that new space. Intuitively, kernel functions provide a similarity relationship between objects being processed, so they are widely also used in similarity-based learning. In this work, we use kernel functions with a twofold purpose: first, to model a more

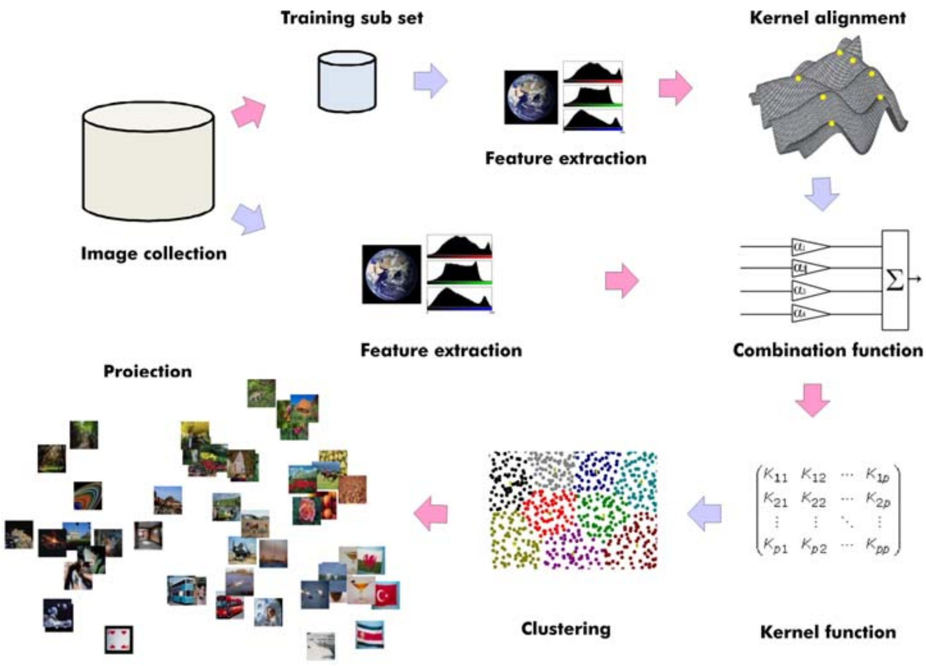


Fig. 1. Framework for summarizing an image collection

appropriate similarity measure between images using low-level visual features, and second, to learn a combination of features adapted to those particularities of the application domain.

A histogram is a discrete and non-parametric representation of a probability distribution function. Although histograms may be seen as feature vectors, they have particular properties that may be exploited by a similarity function. There are different kernel functions specially tailored to histograms. In this work, we use the *histogram intersection kernel*. Consider h as a histogram with n bins, associated to one of different visual features. The histogram intersection kernel between two histograms is defined as $k_{\cap}(h_i, h_j) = \sum_{l=1}^{n} min\,(h_i(l), h_j(l))$. This kernel calculates the area of the intersection of both histograms.

In this work, four different histograms are calculated for each image: borders, texture, RGB and gray. Using $k_{\cap}$ and these four visual features, we obtain four different kernel functions that will be used for building a new kernel. A kernel function using just one low-level feature provides a similarity notion based on particular aspect of the visual perception. For instance, the RGB histogram feature is able to indicate whether two images have similar color distributions. However, we aim to design a kernel function that provides a better notion of image similarity according to the a priori information. We construct the new kernel function using a linear combination of kernel functions associated to individual features. The most simple combination is obtained by assigning equal weights to all basis kernel functions, so the new kernel induces a representation space with all visual features. However, depending on the particular class, some features may have more or less importance.

Cristianini [8] proposed the kernel alignment strategy in the context of supervised learning to combine different visual features in an optimal way with respect to a domain knowledge target (*ideal kernel*). The empirical kernel alignment, is a similarity measure between two kernel functions, calculated over a data sample. If K_α and K_t are the kernel matrices associated to kernel functions k_α and k_t in a data sample S, the kernel alignment measure is defined as:

$$A_S(K_\alpha, K_t) = \frac{\langle K_\alpha, K_t \rangle_F}{\sqrt{\langle K_\alpha, K_\alpha \rangle_F}\sqrt{\langle K_t, K_t \rangle_F}}, \tag{1}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product defined as $\langle A, B \rangle_F = \sum_i \sum_j A_{ij} B_{ij}$, K_α is the linear combination of basis kernels, that is, the combination of all visual features given by $k_\alpha(x, y) = \sum_f \alpha_f k_\cap (h_f(x), h_f(y))$, where $h_f(x)$ is the f-th feature histogram of image x, and α is a weighting vector. The definition of a target kernel function K_t, i.e. an ideal kernel with explicit domain knowledge, is done using labels associated to each image that are extracted from previous information (class labels). It is given by the explicit classification of images for a particular class using y_n as the labels vector associated to the n-th class, in which $y_n(x) = 1$ if the image x is an example of the n-th class and $y_n(x) = -1$ otherwise. So, $K_t = yy'$ is the kernel matrix associated to the target for a particular class. This configuration only considers a two-class case. We need to build a new kernel function that takes into account the information of all classes simultaneously (*multi-class* case).

Vert [12] proposes a strategy that addresses the multi-class problem in the context of multi-class classification. Therefore, we adapted his strategy in the context of image collection summarization. Author proposes to build the ideal kernel matrix as follows:

$$K_t(x, x) = \begin{cases} 1 & \textit{if} \text{ y=y'} \\ -1/(Q-1) & \text{otherwise} \end{cases}, \tag{2}$$

where Q is the number of classes. K_t is, by construction, a valid kernel and we will call it the *ideal kernel*. Under some regularity assumptions on K_α, the alignment function is differentiable with respect to α. Upon this assumption we can use a *gradient ascent* algorithm in order to maximize the alignment between the combined kernel and the ideal kernel as follows:

$$\alpha^* = \underset{\alpha \in \Theta}{argmax}\, A_S(K_\alpha, K_t) \tag{3}$$

Due to the fact that we have a function composed of a vector of variables, we have a *gradient vector* composed of partial derivatives $\nabla \alpha A_S = \left[\frac{\partial A_S}{\partial \alpha_1}, \frac{\partial A_S}{\partial \alpha_2}, \ldots, \frac{\partial A_S}{\partial \alpha_d} \right]^T$. The optimization algorithm starts from a random α, and at each step, updates α, in the direction of the gradient $\Delta\alpha_i = \eta \frac{\partial A_s}{\partial \alpha_i}$, $\forall_i$ and $\alpha_i = \alpha_i + \Delta\alpha_i$, where η is called the *stepsize*, or *learning factor* and determines how much to move in that direction [1].

Kernel alignment strategy has been used in the context of supervised learning and in classification problems. We use it for both, supervised and non-supervised learning in the context of summarization of collection of images.

4 Experimentation

Our main goal in this experimentation was to measure the quality of the summary. We used the Corel data set, which is a collection of photographic stock images and clip art, and it is the most widely used standard data set for testing content based image retrieval systems CBIR. A subset of 2,500 images was selected, which has 25 classes with 100 images each one (aviation, beach, cats, cards, birds, flags, forest, among others). The extracted features were Gray Histogram, RGB color histogram, Sobel Histogram (borders) and Local Binary Partitions (texture). These four visual features were modeled as discrete probability distributions and the kernel function chosen was the histogram intersection. The summary was created with a k-medoids clustering algorithm. For projecting (2D visualization), the original high-dimensional space of the image summary was projected in a low-dimensional space using Multidimensional Scaling (MDS) [11].

4.1 Summary Quality Evaluation

A good summary corresponds to representative set of samples from the collection, i.e., a set that includes prototypical images from the different categories present in the collection. Based on this idea, we define a supervised summarization quality measure that makes use of the image labels. This measure corresponds to the entropy of the summarization and is calculated as follows:

$$H_{summary} = -\sum_{i=1}^{C} (\frac{\#C_i}{k}) log_2(\frac{\#C_i}{k}), \tag{4}$$

where C is the number of classes, $M = \{m_1, \ldots, m_k\}$ is the set of k medoids obtained in the clustering process, and $\#C_i = \|\{m_j \in M | m_j \in C_i\}\|$ is the number of medoids in M that belong to class C_i. The quantity $\frac{\#C_i}{k}$ represents the proportion of samples in the summary that belongs to class C_i. The maximum entropy is obtained when this value is the same for all classes, e.i., $\forall i$, $\frac{\#C_i}{k} = \frac{1}{C}$. In this case, all the classes are equally represented in the summary. The maximum entropy depends on the number of classes, $H_{summary} = log_2(C)$. In this experimental setup $log_2(C) = log_2(25) = 4.64385619$. With this measure defined, we aim to assess the quality of the summaries generated for the following kernel functions: an *ideal kernel function* using the Equation 2, which will have the maximum entropy since it has the a priori class labels information; a *basis kernel function* as a combination of the base kernel functions (RGB, Sobel, LBP and Gray) with equal weights (alphas); and the *aligned kernel* built as was suggested in Section 3.

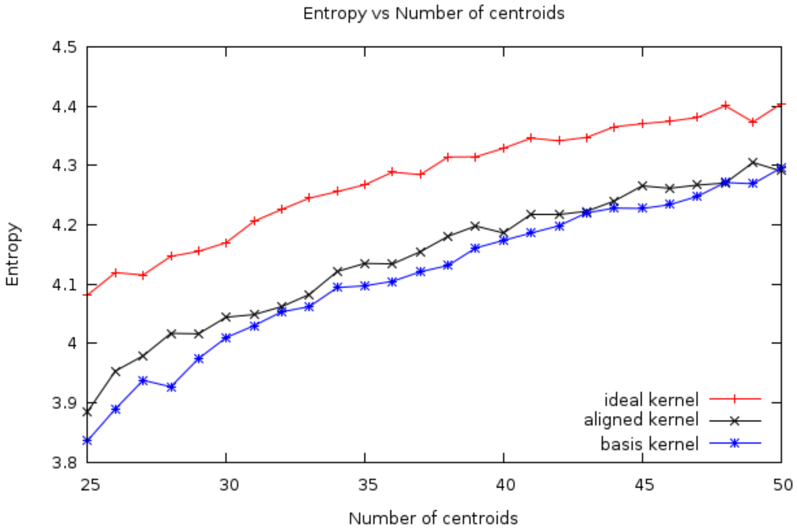


Fig. 2. Entropy vs number of centroids (average for 100 runs). The kernel that involves domain knowledge (*aligned kernel*) outperforms the base kernel.

4.2 Experimental Results

For learning the kernel function, we start the gradient ascent algorithm with α values (one per visual feature) generated randomly (50 times), $\eta = 0.1$, $\nabla\alpha_i = 0.1$ and 100 iterations. Table 1 shows the α values obtained using gradient ascent for optimizing the kernel alignment, which indicates that color feature (RGB) has the highest weight in the combination function. It is because images of the Corel data set have similar color distribution in each class (in other data sets it would be different). On the other hand, texture feature (LBP) has the lowest weight, which indicates that texture is not a good class discriminant in this data set. Figure 2 shows the quality of the three summaries: *ideal kernel, basis kernel,* and *aligned kernel.* Results show that the *aligned kerne*l outperforms the baseline, which proves that the proposed method improves the quality of the summary. All three kernel increase the summary entropy when the number of medoids is increased; with $k{=}50$ medoids the summary entropy is close to the maximum. Figure 3 shows the visualization (2D projection) of the entire Corel data set using MDS with the medoids of each cluster. Figure 4 shows a visualization of the the summary, which involves

Table 1. Alpha values found for the combination function obtained with multi-class kernel alignment

Feature	GRAY	LBP	SOBEL	RGB
α	0.1537	0.0507	0.1023	0.6932

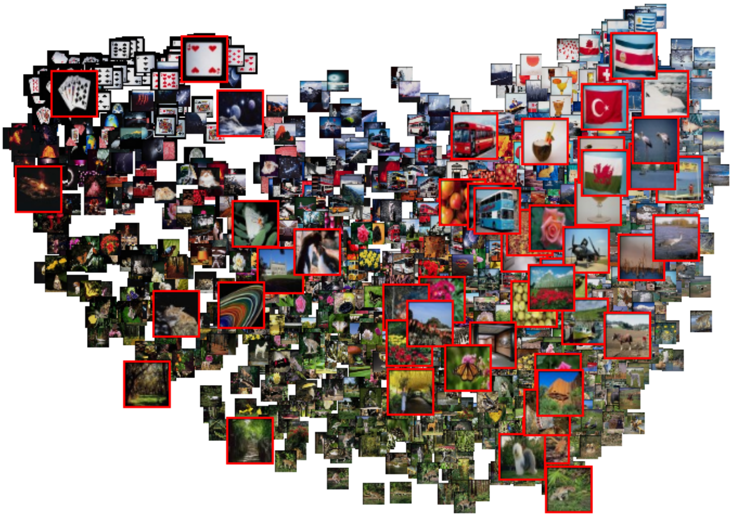


Fig. 3. Visualization of the Corel collection with 50 medoids highlighted

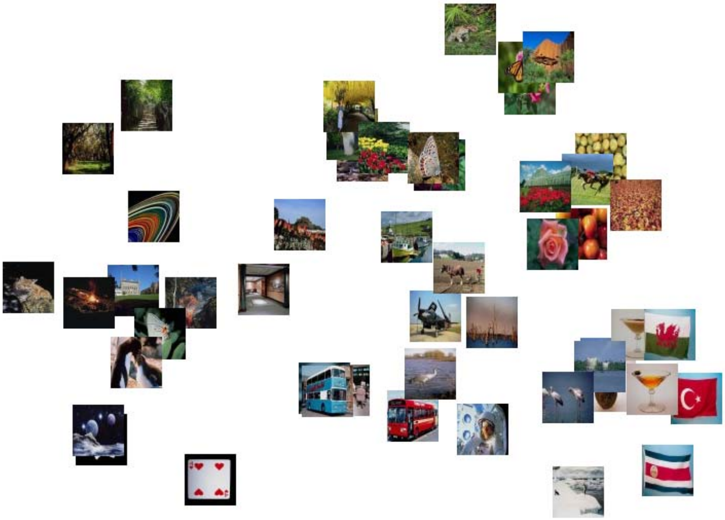


Fig. 4. Visualization of the image collection summary

domain knowledge and represents the entire collection with a higher precision than a summary without domain knowledge.

5 Conclusions and Future Work

We have presented a method for involving domain knowledge in the construction of image collection summaries. We use kernel alignment strategy for both, supervised and non-supervised learning in the context of summarization of large collections of images. We propose a quantitative measure based on information theory for assessing the quality of the summary. Results show that the summary is improved when it is built following the proposed method. With the model proposed in this work, it is possible to automatically classify a new image that arrives to the collection in one of the clusters by calculating the combination function for the new image and calculating its similarity with respect to the images of the summary. In future work, we will evaluate other clustering techniques and we will assess the strategy with real users using quantitative and qualitative measures that allow us to fit the model.

References

1. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
2. Cai, D., He, X., Li, Z., Ma, W.-Y., Wen, J.-R.: Hierarchical clustering of www image search results using visual, textual and link information. In: Proceedings of the 12th annual ACM international conference on Multimedia, pp. 952–959 (2004)
3. Chen, J.-Y., Bouman, C.A., Dalton, J.C.: Hierarchical browsing and search of large image databases. IEEE Transactions on Image Processing 9(3), 442–455 (2000)
4. Joshi, D., Li, J., Wang, J.Z., Datta, R.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40(2), 1–60 (2008)
5. Deng, D.: Content-based image collection summarization and comparison using self-organizing maps. Pattern Recognition 40(2), 718–727 (2007)
6. Gao, B., Liu, T.-Y., Qin, T., Zheng, X., Cheng, Q.-S., Ma, W.-Y.: Web image clustering by consistent utilization of visual features and surrounding texts. In: MULTIMEDIA 2005: Proceedings of the 13th annual ACM international conference on Multimedia, pp. 112–121. ACM, New York (2005)
7. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. Journal of Visual Languages & Computing 19(2), 203–224 (2008)
8. Shawe Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
9. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: IEEE 11th International Conference on Computer Vision, 2007 (ICCV 2007), pp. 1–8 (2007)
10. Stan, D., Sethi, I.K.: eid: a system for exploration of image databases. Inf. Process. Manage. 39(3), 335–361 (2003)
11. Torgerson, M.S.: Multidimensional scaling: I. theory and method. Psychometrika 17(4), 401–419 (1958)
12. Vert, R.: Designing a m-svm kernel for protein secondary structure prediction. Master's thesis, DEA informatique de Lorraine (2002)

X Statistical Pattern Recognition

Correlation Pattern Recognition in Nonoverlapping Scene Using a Noisy Reference

Pablo M. Aguilar-González and Vitaly Kober

Department of Computer Science, Centro de Investigación Científica y de Educación Superior de Ensenada,
Km. 107 Carretera Tijuana-Ensenada, Ensenada 22860, B.C., México
{paguilar,vkober}@cicese.mx
http://www.cicese.mx/

Abstract. Correlation filters for recognition of a target in nonoverlapping background noise are proposed. The object to be recognized is given implicitly; that is, it is placed in a noisy reference image at unknown coordinates. For the filters design two performance criteria are used: signal-to-noise ratio and peak-to-output energy. Computer simulations results obtained with the proposed filters are discussed and compared with those of classical correlation filters in terms of discrimination capability.

Keywords: correlation filters, pattern recognition.

1 Introduction

Since the pioneering work by VanderLugt [1], correlation filters have been extensively studied for the purpose of pattern recognition [2-15]. Within the context of pattern recognition, detection and location estimation are two very important tasks. When a correlation filter is used, such tasks may be solved in two steps; that is, first, the detection is carried out by searching the highest correlation peaks at the filter output, and then the coordinates of the peaks are taken as position estimations of targets in the scene image [2].

Different criteria have been proposed to evaluate the performance of correlation filters [3] such as signal-to-noise ratio (SNR), peak sharpness, light efficiency, discrimination capability, etc. Filters are designed by maximizing one of these criteria. Many filters have been proposed when an input scene contains a target distorted by additive noise. The matched filter (MF) [1] is derived by maximizing the SNR. The phase-only filter [4] maximizes light efficiency. The optimal filter (OF) [5] minimizes the probability of anomalous errors. Several filters have been derived for the nonoverlapping scene model [6,7,8,9,10]. The generalized matched filter [7] was derived by maximizing the ratio of the square of the expected value of the correlation peak to the average output variance. The generalized optimum filter [7] maximizes the peak-to-output energy ratio (POE). Recently [11], several correlation filters were proposed for the scene model that takes into account linear degradations of the both scene and target.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 555–562, 2009.

All of these filters, however, are derived under the assumption that a target is explicitly known. However, in real-life situations the target is often given by a reference image, which contains the reference object at unknown coordinates, as well as a noisy background. In a recent paper [12] a signal model was proposed that takes into account additive noise in the reference image to design filters for detecting a target in overlapping noise. The considered signal model is close to practical situations, in which observed and reference images are inevitably corrupted by noise owing to the image formation process. In this paper, extend that work to account for the presence of a nonoverlapping background in the input scene. We propose two correlation filters optimized with respect to the $\tilde{\text{SNR}}$ and POE. The performance of the filters is compared to that of classical correlation filters.

2 Analysis

We use the additive signal model for the reference image and the nonoverlapping signal model for the input scene. For simplicity, 1-dimensional notation is used. Formally, the scene and reference image are given, respectively, by

$$s_o(x) = t(x - x_s) + b(x)\,\bar{w}(x - x_s) + n_s(x) \ , \tag{1}$$

$$r(x) = t(x - x_r) + n_r(x) \ , \tag{2}$$

where $t(x)$ is the target, $s_o(x)$ is the observed scene with the target location x_s, $r(x)$ is the reference image with the target located at the coordinate x_r, and $n_s(x)$ and $n_r(x)$ are noise signals in the input scene and the reference image, respectively. $b(x)$ is the nonoverlapping background, treated as the realization of a stationary random process with the mean μ_s and the power spectral density $B_0(\omega)$, and it is multiplied by $\bar{w}(x)$ the inverse support function of the target. Both $n_s(x)$ and $n_r(x)$ are assumed to be stationary random processes. It is also assumed that the random processes and the random target locations x_s and x_r are statistically independent of each other. We will design filters to be applied to the centered scene, that is $s(x) = s_o(x) - \mu_s$. $S(\omega)$ and $T(\omega)$ are the Fourier transforms of $s(x)$ and $t(x)$, respectively, and $N_s(\omega)$ and $N_r(\omega)$ are the power spectral densities of $n_s(x)$ and $n_r(x)$, respectively.

Since the target signal is not available, we look for a correlation filter of the following form:

$$H(\omega) = A(\omega)\,R^*(\omega) \ , \tag{3}$$

where $A(\omega)$ is a deterministic function, $R(\omega)$ is the Fourier transform of the realization of the reference image given in (2), and * denotes complex conjugate. Actually, it is interesting to note that the filter is given by a bank of the transfer functions determined by a realization of the random process n_r.

Because the location of the target in the reference image is x_r and not the origin, the correlation output peak is expected at the coordinate $x_s - x_r$. Note however that as long as the target is reasonably centered in the reference image, the location estimation of the target in the input scene will be in the vicinity of the true location. Even if the exact location of the target can't be determined, knowing the relative position is useful for applications like tracking [16], where what is important is the tracked object's trajectory.

The modified generalized optimum filter (GOF_{AN}) is derived by maximizing the POE criterion:

$$\text{POE} = \frac{\left|\text{E}\left\{y\left(x_s - x_r\right)\right\}\right|^2}{\text{E}\left\{\overline{\left|y\left(x\right)\right|^2}\right\}} \,, \tag{4}$$

where $\text{E}\left\{\cdot\right\}$ denotes the expected value and and the over-bar denotes spatial averaging, i.e. $\overline{y\left(x\right)} = \left(1/L\right) \int y\left(x\right) dx$, L is the spatial extent of the signal $y\left(x\right)$. The expected filter output at the location of the correlation peak can be calculated as

$$\text{E}\left\{y\left(x_s - x_r\right)\right\} = \frac{1}{2\pi} \int A\left(\omega\right) \text{E}\left\{\left(R\left(\omega\right) e^{i\omega x_r}\right)^* S\left(\omega\right) e^{i\omega x_s}\right\} d\omega \,. \tag{5}$$

We can calculate $\text{E}\left\{\overline{\left|y\left(x\right)\right|^2}\right\}$ as

$$\text{E}\left\{\overline{\left|y\left(x\right)\right|^2}\right\} = \frac{1}{2\pi L} \int \left|A\left(\omega\right)\right|^2 \text{E}\left\{\left|R^*\left(\omega\right) S\left(\omega\right)\right|^2\right\} dx \,. \tag{6}$$

Substituting (5) and (6) into (4) we get

$$\text{POE} = \frac{L \left|\int A\left(\omega\right) \text{E}\left\{\left(R\left(\omega\right) e^{i\omega x_r}\right)^* S\left(\omega\right) e^{i\omega x_s}\right\} d\omega\right|^2}{2\pi \int \left|A\left(\omega\right)\right|^2 \text{E}\left\{\left|R^*\left(\omega\right) S\left(\omega\right)\right|^2\right\} dx} \,, \tag{7}$$

and applying the Cauchy-Schwartz inequality, the expression for the GOF_{AN} is given by

$$\text{GOF}_{\text{AN}}\left(\omega\right) = \frac{\left(\left|T\left(\omega\right)\right|^2 - \mu_s T\left(\omega\right) W^*\left(\omega\right)\right) R^*\left(\omega\right)}{\left(\left|T\left(\omega\right)\right|^2 + N_r\left(\omega\right)\right) \left(\left|T_s\left(\omega\right)\right|^2 + \frac{1}{2\pi} B_0\left(\omega\right) * \left|\bar{W}\left(\omega\right)\right|^2 + N_s\left(\omega\right)\right)} \,, \tag{8}$$

where $T_s\left(\omega\right)$ is the Fourier transform of the expected value of the centered input scene, namely $t_s\left(x\right) = t\left(x\right) - \mu_s w\left(x\right)$. $w\left(x\right)$ is the support function for the target, that is $w\left(x\right) = 1 - \bar{w}\left(x\right)$.

The modified matched filter (GMF_{AN}) can be derived by maximizing the $\text{S}\tilde{\text{N}}\text{R}$ criterion:

$$\text{S}\tilde{\text{N}}\text{R} = \frac{\left|\text{E}\left\{y\left(x_s - x_r\right)\right\}\right|^2}{\overline{\text{Var}\left\{y\left(x\right)\right\}}} \,, \tag{9}$$

where $\text{Var}\left\{\cdot\right\}$ denotes variance. The variance of the output is given by

$$\overline{\text{Var}\left\{y\left(x\right)\right\}} = \frac{1}{2\pi L} \int \left|A\left(\omega\right)\right|^2 \text{Var}\left\{R^*\left(\omega\right) S\left(\omega\right)\right\} d\omega \,. \tag{10}$$

Substituting (5) and (10) into (9) we obtain the following expression for the SNR:

$$\tilde{\mathrm{SNR}} = \frac{L\left|\int A\left(\omega\right)\mathrm{E}\left\{\left(R\left(\omega\right)e^{i\omega x_r}\right)^{*}S\left(\omega\right)e^{i\omega x_s}\right\}d\omega\right|^{2}}{2\pi\int\left|A\left(\omega\right)\right|^{2}\mathrm{Var}\left\{R^{*}\left(\omega\right)S\left(\omega\right)\right\}d\omega}\ . \tag{11}$$

Applying the Cauchy-Schwartz inequality we obtain the expression for the $\mathrm{GMF}_{\mathrm{AN}}$

$$\mathrm{GMF}_{\mathrm{AN}}\left(\omega\right) = \frac{\left(\left|T\left(\omega\right)\right|^{2}-\mu_{s}T\left(\omega\right)W^{*}\left(\omega\right)\right)R^{*}\left(\omega\right)}{\left(\left|T\left(\omega\right)\right|^{2}+N_{r}\left(\omega\right)\right)\left(\left|T_{s}\left(\omega\right)\right|^{2}+\frac{1}{2\pi}B_{0}\left(\omega\right)*\left|\bar{W}\left(\omega\right)\right|^{2}+N_{s}\left(\omega\right)\right)-\left|T\left(\omega\right)T_{s}\left(\omega\right)\right|^{2}}\ . \tag{12}$$

Note that if the reference image does not contain noise, the $\mathrm{GMF}_{\mathrm{AN}}$ and the $\mathrm{GOF}_{\mathrm{AN}}$ are equal to the classical GMF and GOF, respectively.

It can be seen that the obtained filters require knowledge of the target Fourier transform, and the support function. This contradicts the assumption that information about the target is unknown. However, estimations may be designed from the available information. We can apply the smoothing Wiener filter [17] to attenuate the effects of noise in the reference image. After that, we can apply a threshold to the resulting image and obtain an approximate support function.

$$\tilde{r}\left(x\right) = r\left(x\right)*h_{wiener}\left(x\right)\ , \tag{13}$$

$$\tilde{w}\left(x\right) = \begin{cases}1 & \tilde{r}\left(x\right)\geq\tau\left(\mu_{t},\,\sigma_{t},\,\sigma_{n_r}\right)\\ 0 & \text{otherwise}\end{cases}\ . \tag{14}$$

The optimum threshold depends on the statistics of the input noise and the target. When the target mean and standard deviation are known, these values can be used to improve the threshold selection. If such values are unknown, the optimum threshold can be determined in terms of the input noise statistics. If the support function estimation is not reliable owing to the presence of high levels of noise in the reference image, we can approximate the optimum filter transfer function by disregarding terms that require the support function. Figure 1 shows the regions of the parameter space in which each estimation is best. When SNR is high, we can correctly estimate the support function and use that with the original reference image. When SNR is not high enough, it is better to use the Wiener filtered reference image. And in some cases, when the input SNR is low and the target occupies a large percent of the reference image, it is better to not consider the estimated support function because it introduces errors inside the area occupied by the target. In these cases the noise present outside of the target is small and affects performance less than an incorrect estimation of the support function. Target image estimations based on the three regions are given as follows:

$$\tilde{t}\left(x\right) = \begin{cases}\tilde{r}\left(x\right)\ , & \text{if input SNR is low and target is large}\\ \tilde{r}\left(x\right)\tilde{w}\left(x\right)\ , & \text{if input SNR is low and target is small}\\ r\left(x\right)\tilde{w}\left(x\right)\ , & \text{if input SNR is high}\end{cases}\ . \tag{15}$$

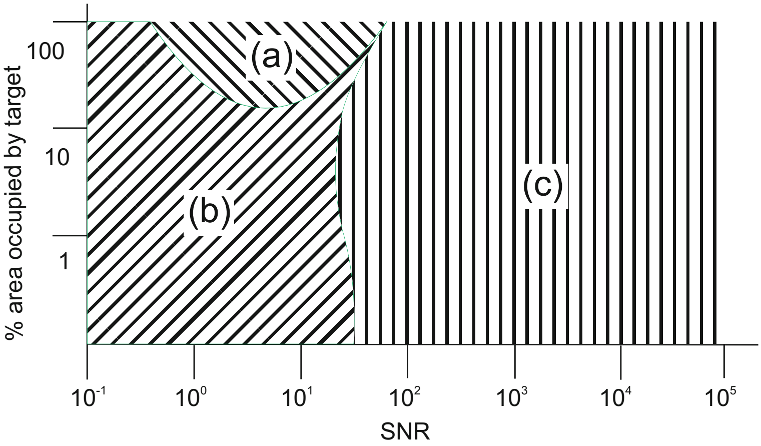


Fig. 1. Regions of the SNR-Area parameter space where each estimation is best: (a) better not to estimate the support function, (b) better to estimate target using filtered reference image, and (c) better to estimate target using estimated support function and the original reference image

3 Computer Simulations

In this section we present computer simulation results. The performance of the proposed filters are compared with that of the classical filters in terms of the discrimination capability (DC). The DC is formally defined [5] as the ability of a filter to distinguish a target from other objects in the scene. The DC can be expressed as follows:

$$DC = 1 - \frac{\left|C^{B}(0)\right|^{2}}{\left|C^{T}(0)\right|^{2}}, \tag{16}$$

where $\left|C^{B}(0)\right|$ is the maximum value in the correlation plane over the background area, and $\left|C^{T}(0)\right|$ is the maximum value in the correlation plane over the area the target occupies in the input scene. The background area and the target area are complementary. Values of the DC close to unity, indicate a good capacity to discriminate the target against unwanted objects. Negative values of the DC indicate a failure to detect the target. We show simulation results when using generalized optimum filters only. Generalized matched filters do not control the output mean value which may result in a correlation peak being buried in output noise that has a high mean [7] and thus perform poorly in terms of DC.

The size of all images used in the experiments is 256 × 256 pixels. All filters are implemented using the Discrete Fourier Transform. The intensity values are in the range [0–255]. We use the butterfly shown in Fig. 2(a) as a target. There are two background types, shown in Fig. 2: deterministic and stochastic backgrounds. The stochastic background is a realization of a colored random process with the mean and standard deviation of 115 and 40, respectively, and with the horizontal and vertical correlation coefficients of 0.95. To guarantee statistically correct results, 30 statistical trials of each experiment for either different positions of the target or realizations of random processes were performed. The sample reference images are corrupted by additive white Gaussian noise.

Three filters are used in the experiments: the classical GOF which is designed with all parameters known to establish an upper bound on performance; the proposed $\mathrm{GOF}_{\mathrm{AN}}$ filter when estimating the target as $r\left(x\right)\tilde{w}\left(x\right)$, shown as $\mathrm{GOF}_{\mathrm{AN}}1$ in the simulation results; and the proposed filter when estimating the target as $\tilde{r}\left(x\right)\tilde{w}\left(x\right)$, shown as $\mathrm{GOF}_{\mathrm{AN}}2$.

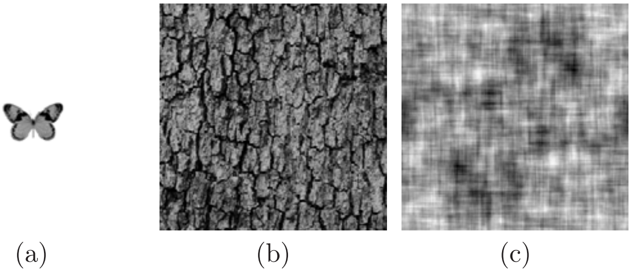


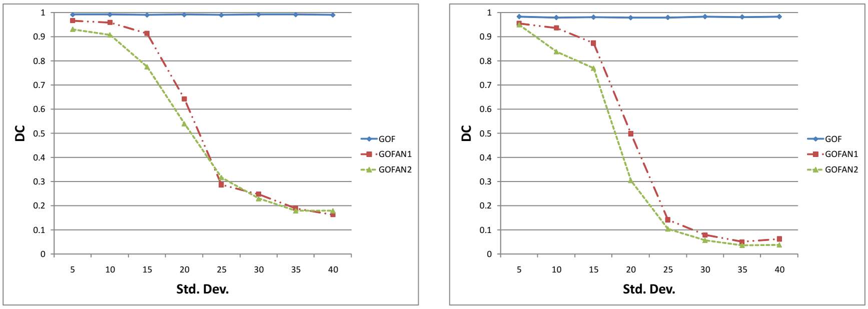
Fig. 2. (a) The target used in the experiments, (b) deterministic background, (c) example of stochastic background

3.1 Scenario 1: Stochastic Background

In order to determine the performance of the proposed filters, we performed experiments for different realizations of the background process while the location of the target within the scene was varied. The simulation results are shown in Fig. 3. It can be seen that the performance for the GOF remains constant. It is because this filter is designed with all parameters known and no noise presence. It can also be seen that the proposed filters are able to detect the target even in the presence of noise of a Std. Dev. of up to 20. When there are higher levels of noise in the reference image performance drops quickly. This is because we are unable to estimate the target support function and we also can not design the filter by ignoring the support function, because the target information is mostly destroyed by the presence of noise. It can be seen that performance behaves similarly regardless of the level of noise present in the input scene.

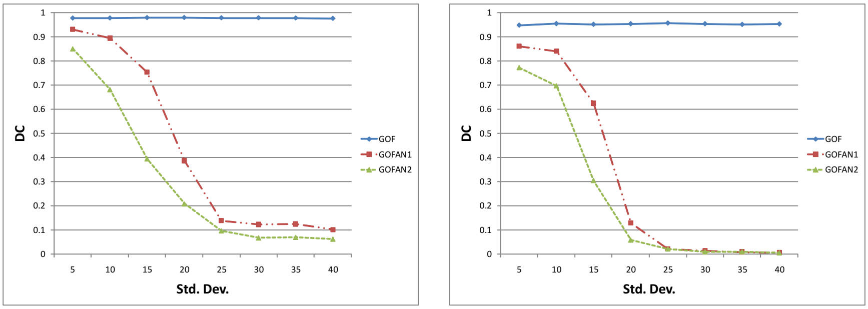
3.2 Scenario 2: Deterministic Background

We also test the performance of the proposed filters in when the background is a deterministic scene. The results are shown in Fig. 4. In this scenario the performance of the GOF also remains constant because it is not affected by the reference image noise. For the proposed filters, the mean value of the DC decreases as the noise in the reference image surpasses the same threshold as in



(a) (b)

Fig. 3. Performance of correlation filters in terms of DC while varying the Std. Dev. of the reference image noise. The scene has a stochastic background and additive scene noise Std. Dev. of (a) 5 and (b) 15.



(a) (b)

Fig. 4. Performance of correlation filters in terms of DC while varying the Std. Dev. of the reference image noise. The scene has a deterministic background and additive scene noise Std. Dev. of (a) 5 and (b) 15.

scenario 1. Detection performance decreases more as the additive noise in the input scene increases but the behavior remains the same.

4 Conclusion

In this paper new correlation filters for recognition of a target in nonoverlapping background noise were proposed. The filters are derived from a new reference model, which takes into account the presence of additive noise in the reference image. With the help of computer simulations, we showed that the proposed filters yield good results in the presence of moderate levels of noise. It was also shown that the proposed filters are robust to different realizations of the reference image noise.

References

1. VanderLugt, A.B.: Signal Detection by Complex Filtering. IEEE Trans. Inf. Theory IT 10, 139–145 (1964)
2. Vijaya-Kumar, B.V.K., Mahalanobis, A., Juday, R.D.: Correlation Pattern Recognition. Cambridge University Press, Cambridge (2005)
3. Vijaya-Kumar, B.V.K., Hassebrook, L.: Performance Measures for Correlation Filters. Appl. Opt. 29, 2997–3006 (1990)
4. Horner, J.L., Gianino, P.D.: Phase-Only Matched Filtering. Appl. Opt. 23, 812–816 (1984)
5. Yaroslavsky, L.P.: The Theory of Optimal Methods for Localization of Objects in Pictures. In: Wolf, E. (ed.) Progress in Optics, vol. 33, pp. 145–201. Elsevier, Amsterdam (1993)
6. Javidi, B., Wang, L.: Optimum Filter for Detection of a Target in Nonoverlapping Scene Noise. Appl. Opt. 33, 4454–4458 (1994)
7. Javidi, B., Wang, J.: Design of Filters to Detect a Noisy Target in Nonoverlapping Background Noise. J. Opt. Soc. Am. A 11, 2604–2612 (1994)
8. Réfrégier, P., Javidi, B., Zhang, G.: Minimum Mean Square Error Filter for Pattern Recognition With Spatially Disjoint Signal and Scene Noise. Opt. 18, 1453–1455 (1993)
9. Javidi, B., Horner, J.L.: Real-Time Optical Information Processing. Academic Press, Inc., London (1994)
10. Kober, V., Campos, J.: Accuracy of Location Measurement of a Noisy Target in a Nonoverlapping Background. Journal OSA 13, 1653–1666 (1996)
11. Ramos, E.M., Kober, V.: Design of Correlation Filters for Recognition of Linearly Distorted Objects in Linearly Degraded Scenes. J. Opt. Soc. Am. A 24, 3403–3417 (2007)
12. Aguilar-González, P.M., Kober, V.: Correlation Filters for Pattern Recognition Using a Noisy Reference. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 38–45. Springer, Heidelberg (2008)
13. Kober, V., Ovseyevich, A.: Phase-Only Filter with Improved Filter Efficiency and Correlation Discrimination. Pattern Recognition and Image Analysis 10, 514–519 (2000)
14. Díaz-Ramírez, V.H., Kober, V., Álvarez-Borrego, J.: Pattern Recognition With an Adaptive Joint Transform Correlator. Appl. Opt. 45, 5929–5941 (2006)
15. González-Fraga, J.A., Kober, V., Álvarez-Borrego, J.: Adaptive Synthetic Discriminant Function Filters for Pattern Recognition. Opt. Eng. 45, 1–10 (2006)
16. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38(4), Article 13 (2006)
17. Pratt, W.K.: Digital Image Processing. John Wiley & Sons, Chichester (2001)

Particle Swarm Model Selection for Authorship Verification

Hugo Jair Escalante, Manuel Montes, and Luis Villaseñor

Laboratorio de Tecnologías del Lenguaje
INAOE, Luis Enrique Erro No. 1, 72840, Puebla, México
{hugojair,mmontesg,villasen}@inaoep.mx

Abstract. Authorship verification is the task of determining whether documents were or were not written by a certain author. The problem has been faced by using binary classifiers, one per author, that make individual yes/no decisions about the authorship condition of documents. Traditionally, the same learning algorithm is used when building the classifiers of the considered authors. However, the individual problems that such classifiers face are different for distinct authors, thus using a single algorithm may lead to unsatisfactory results. This paper describes the application of particle swarm model selection (PSMS) to the problem of authorship verification. PSMS selects an ad-hoc classifier for each author in a fully automatic way; additionally, PSMS also chooses preprocessing and feature selection methods. Experimental results on two collections give evidence that classifiers selected with PSMS are advantageous over selecting the same classifier for all of the authors involved.

1 Introduction

Author verification (AV) is the task of deciding whether given text documents were or were not written by a certain author [13]. There is a wide field of application for this sort of methods, including spam filtering, fraud detection, computer forensics and plagiarism detection. In all of these domains, the goal is to confirm or reject the authorship condition for documents with respect to a set of candidate authors, given sample documents written by the considered authors. In the past decade this task was confined to stylography experts who should analyze sample texts from authors to make a decision about the authorship of documents. However, the increasing demand for AV techniques and its wide scope of application have provoked an increasing interest on the scientific community for developing automatic methods for AV.

The scenario we consider is as follows. For each author, we are given sample documents[1] written by her/him as well as documents written by other authors. Features are extracted from documents for representing them in an amenable way for statistical modeling, a model is then built (based on the derived representations for documents) for the author. When a new document arrives, the

[1] We consider digital text documents only, although the proposed methods can be applied to other type of documents (e.g., scanned handwritten documents) as well.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 563–570, 2009.

model must be able to decide whether the document was written by the author or not. Thus, the AV task can be posed as one of binary classification, with a classifier per author. Under this setting, sample documents for the author under consideration are considered positive examples, whereas sample documents for other authors are considered negative examples.

Usually, the same learning algorithm is used to build all of the classifiers corresponding to the set of considered authors. However, the individual problem that each classifier faces is different for distinct authors, thus there is no guarantee that using the same algorithm for all of the authors would lead to acceptable results. Also, while some features may be useful for building the classifier for author *"X"*, the same features may be useless for modeling author *"Y"*. Thus, whenever possible, specific features and classifiers should be considered for different authors. Unfortunately, manually selecting specific features and classifiers for each author is impractical and thus we must resort to automatic techniques.

This paper describes the use of particle swarm model selection (PSMS) for the problem of authorship verification. PSMS can select an *ad-hoc* classifier for each author in a fully automatic way; additionally, PSMS also chooses specific preprocessing and feature selection methods for each problem. This formulation allows us to model each author independently, which results in a more reliable modeling and hence in better verification performance. We conducted experiments on two collections comprising different numbers of authors, samples, lengths of documents and languages, which allows us evaluating the generality of the formulation. Experimental results give evidence that classifiers selected with PSMS are advantageous over selecting the same classifier for all of the authors involved. Also, the methods selected with PSMS can be helpful to gain insight into the distinctive features associated to authors. The rest of this paper describes related work on AV (Section 2); our approach to AV based on PSMS (Section 3); experimental results (Section 4) that show the relevance of the proposed formulation; and the conclusions derived from this work (Section 5).

2 Related Work

Most of the work on AV has focused on developing specific features (stylometric, lexical, character-level, syntactic, semantic) able to characterize the writing style of authors, thus putting emphasis on feature extraction and selection [7,11,10,1], see [13] for a comprehensive review. However, despite these features can be helpful for obtaining reliable models, extracting such features from raw text is a rather complex and time consuming process. In contrast, in this work we adopt a simple set of features to represent the documents and focus on the development of reliable classification models.

The AV problem has been formulated either as a one-class classification problem or as a one-vs-all multiclass classification task. In the former case, sample documents are available from a single author [10] (Did author *"X"* write the document or it was written by any other author?), while in the second case, samples are available from a set of candidate authors [7,11,1] (give the most probable candidate from a list of authors). This paper adopts the second formulation as it is a more controlled and practical scenario. Those works that have adopted the one-vs-all paradigm consider as positive examples to documents written by an author and negative examples to documents written by the rest of the candidate authors. Then, binary classifiers are built such that they are able to determine whether unseen texts have been written by an author or not.

To the best of our knowledge, all of the reported methods adopting this formulation have used the same learning algorithm to build the classifiers for different authors [7,11,1]. However, using the same learning method for all of the authors does not guarantee that the individual models are the best ones for each author. Also, most of the related works have used the same preprocessing processes and features for all of the authors. The latter leads to obtain consistent outputs across different classifiers, which can be helpful for authorship attribution. Nevertheless, the individual modeling will not be as effective as if we consider specific methods for each author. For that reason, in this work we propose using particular models for each of the authors under consideration.

Model selection is the task of selecting the best model for classification given a set of candidates [8]. Traditionally, a single learning algorithm is considered and the task is to optimize the model's parameters such that its generalization performance is maximized [12]. A problem with most model selection techniques is that they require users to provide prior domain-knowledge or to supply preprocessed data in order to obtain reliable models [6]. PSMS is a more ambitious formulation that selects full models for classification without requiring much supervision [4]. Only a few methods have been proposed for facing the *full model selection* problem, most notably the work by Gorissen et al. [5]. Unlike the latter method, PSMS is more efficient and simple to implement, moreover, PSMS has shown to be robust against overfitting because of the way the search is guided.

3 Particle Swarm Model Selection for Author Verification

Our approach to AV follows the standard scenario described in Section 1, using PSMS for constructing the model for each author. Specifically, we are given N sample documents, each written by one of M authors. Each document d^i is represented by its bag-of-words, $\mathbf{v}^i \in [0,1]^{|V|}$, which is a boolean vector of the size of the collection's vocabulary V; each entry j in $\mathbf{v}^i_j$ indicates whether word $w_j \in V$ appears in document d^i or not. We build M training data sets for binary classification considering the bags-of-words of the N samples and assigning labels to training examples in a different way for each data set. For each author $C_i \in \{C_1, \dots C_M\}$ we build a data set D_i such that we assign the positive label $(+1)$ to documents written by author C_i and the negative one (-1) to documents written by other authors $C_{j:j \neq i}$. Thus we obtain M training sets, each of the form $D_i = \{(\mathbf{v}^1, l^1), \dots, (\mathbf{v}^N, l^N)\}$, with $l^i \in \{-1, 1\}$. At this stage we apply PSMS to select a specific classification model for each author, using the corresponding data sets. Besides classifier, PSMS selects methods for preprocessing and feature selection, and optimizes hyperparameters for the selected methods. The model selected with PSMS is trained using the available data and tested in a separate test set. The rest of this section describes the PSMS technique.

3.1 Particle Swarm Model Selection

PSMS is the application of Particle swarm optimization (PSO) to the model selection problem in binary classification [4]. Given a machine learning toolbox PSMS selects the best combination of methods for preprocessing, feature selection and classification. Additionally, PSMS optimizes hyperparameters of the selected methods. PSMS explores the classifiers space by means of PSO, which optimizes the classification error using training data; as PSO searches both locally and globally, it allows PSMS to overcome, to some extent, overfitting [4].

PSO is a bio-inspired search technique that has proved to be very effective in several domains [3]. The algorithm mimics the behavior of biological societies that share goals and present local and social behavior. Solutions are called particles, at each iteration t, each particle has a position in the search space $\mathbf{x}_i^t =< x_{i,1}^t, \ldots, x_{i,d}^t >$, and a velocity $\mathbf{v}_i^t =< v_{i,1}^t, \ldots, v_{i,d}^t >$ value, with d the dimensionality of the problem. The particles are randomly initialized and iteratively update their positions in the search space as follows $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}$, with $\mathbf{v}_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (\mathbf{p}_i - \mathbf{x}_i^t) + c_2 \times r_2 \times (\mathbf{g}_i - \mathbf{x}_i^t)$; where $\mathbf{p}_i$ is the best position obtained by $\mathbf{x}_i$, $\mathbf{g}_i$ is the best particle in the swarm, c_1 and c_2 constants and r_1, r_2 random numbers, w is the so called inertia weight, see [3] for details. The goodness of particles is evaluated with a fitness function specific for the task at hand. PSO stops when a fixed number of iterations is performed.

In PSMS the particles are full models (i.e., combinations of preprocessing, feature selection and classification methods), codified as numerical vectors. The optimization problem is minimizing an estimate of classification error. In particular, we consider the balanced error rate (BER) as fitness function; $BER = \frac{E_+ + E_-}{2}$, where E_+ and E_- are the error rates in the positive and negative classes, respectively. As the test data are unseen during training, the error of solutions (i.e., full models) is estimated with $k-$fold cross validation (CV) on the training set. Thus, the PSO algorithm is used to search for the model that minimizes the CV-BER. The selected model is considered the classifier for the corresponding author in AV. We consider the PSMS implementation included in the CLOP[2] toolbox. Table 1 shows the methods from which PSMS can choose, see [4] for further details. PSMS has reported outstanding results on diverse binary classification problems without requiring significant supervision [6,4], which makes it attractive for many applications. The application of PSMS to AV arises naturally, as we want to select specific full models for each author.

4 Experimental Results

We report results on two collections described in Table 2. The collections have heterogeneous characteristics which make them particularly useful to test the robustness of PSMS to different training set sizes, dimensionality, languages and number of authors. Both collections have predefined partitions for training/testing that have been used in previous works for authorship attribution [2,9]. We kept the

[2] http://clopinet.com/CLOP

Table 1. Classification (C), feature selection (F) and preprocessing (P) methods considered in our experiments; we show the object name and the number of parameters for each method

Object name	Type	# pars.	Description
zarbi	C	0	Linear classifier
naive	C	0	Naïve Bayes
logitboost	C	3	Boosting with trees
neural	C	4	Neural network
svc	C	4	SVM classifier
kridge	C	4	Kernel ridge regression
rf	C	3	Random forest
lssvm	C	5	Kernel ridge regression
Ftest	F	4	F-test criterion
Ttest	F	4	T-test criterion
aucfs	F	4	AUC criterion
odds-ratio	F	4	Odds ratio criterion
relief	F	3	Relief ranking criterion
Pearson	F	4	Pearson correlation coefficient
ZFilter	F	2	Statistical filter
s2n	F	2	Signal-to-noise ratio
$pc - extract$	F	1	Principal components analysis
svcrfe	F	1	SVC- recursive feature elimination
normalize	P	1	Data normalization
standardize	P	1	Data standardization
$shift - scale$	P	1	Data scaling

Table 2. Collections considered for experimentation

Collection	Training	Testing	Features	Authors	Language	Domain	Ref.
MX-PO	281	72	8,970	5	Spanish	Poetry	[2]
CCAT	2,500	2,500	3,400	50	English	News	[9]

words that appear at least in 5 and 20 documents, for the MX-PO and CCAT collections, respectively. We report average precision (P) and recall (R), as well as the F_1 measure, defined as $F_1 = \frac{2 \times R \times P}{R+P}$, and the BER of the individual classifiers.

Besides applying PSMS as described in Section 3.1 (see **FMS/1** below), we investigate the usefulness of PSMS under two other settings that have not been tested elsewhere. This is with the goal of evaluating the benefits of introducing prior knowledge provided by the user. The considered settings are as follows:

- **FMS/1** selects preprocessing, feature selection and classification methods.
- **FMS/0** selects preprocessing and feature selection methods only.
- **FMS/-1** hyperparameter optimization for a fixed classifier.

Through settings **FMS/0** and **FMS/-1**, the user provides prior knowledge by fixing a classification method. Therefore, better results are expected with these settings. Besides using PSMS for the selection of classifiers, we also used the classifiers shown in Table 1 with default parameters for comparison.

Table 3 shows the average BER and the F_1 measure obtained by methods we tried for both collections. For the **FMS/0** configuration we fixed the classifier to be *zarbi* for both collections, as this algorithm has no hyperparameters to optimize and thus PSMS would be restricted to search for preprocessing and feature selection methods. For **FMS/-1** we tried different configurations, although the best results were obtained by fixing *neural* and *svc* classifiers for CCAT and MX-PO, respectively.

From Table 3, we can see that classifiers selected with PSMS show better performance than the individual methods. Interestingly, the best results were obtained with the **FMS/0** configuration. Note that we fixed a non-parametric classifier and PSMS selected for preprocessing and feature selection methods. Thus, despite the individual performance of *zarbi* is low, its performance after selecting appropriate methods for preprocessing and feature selection is significantly improved. The performances of the **FMS/1** and **FMS/-1** settings are competitive as well outperforming most of the individual classifiers for both collections. Therefore, in absence of any knowledge about the behavior of the available classifiers it is recommended to use PSMS instead of trying several classifiers and combinations of methods for preprocessing and feature selection.

Table 3. Average BER and F_1-measure for the considered methods in both collections

Col./Clas.	**zarbi**	**naïve**	**lboost**	**neural**	**svc**	**kridge**	**rf**	**lssvm**	**FMS/-1**	**FMS/0**	**FMS/1**
						BER					
MX-PO	34.64	30.24	29.08	28.59	30.81	31.90	48.01	33.52	26.18	**23.68**	26.88
CCAT	14.24	26.21	15.12	41.50	29.18	27.69	47.01	36.64	35.34	**13.54**	16.39
						F_1					
MX-PO	46.26	52.93	53.18	59.25	54.57	52.52	6.66	48.76	58.28	**60.37**	57.09
CCAT	59.69	55.73	47.11	28.46	56.46	51.85	10.58	38.54	44.11	61.17	**63.41**

Table 4 shows the models selected by PSMS under the **FMS/1** configuration for the MX-PO data set. We can see the variety of methods selected, which are different for each author. The BER of the first three authors is below the mean of individual classifiers, while the BER of models for the last two authors is high, even when non-linear models are used for the latter. This suggest that R. Castellanos and R. Bonifaz are more complex to model, and that better features may be needed for building the respective classifiers.

Table 4. Full models selected by PSMS, under **FMS/1**, for the MX-PO collection

Poet	**Preprocessing**	**Feature Selection**	**Classifier**	**BER**
E. Huerta	standardize(1)	-	zarbi	10.28
S. Sabines	-	-	zarbi	26.79
O. Paz	normalize(0)	Zfilter(3070,0.56)	zarbi	25.09
R. Castellanos	normalize(0)	Zfilter(7121,0.001)	kridge(rbf-γ =0.45)	33.04
R. Bonifaz	shift-scale(1)	-	neural(u=3;iter=15)	35.71

Table 5 shows statistics on the selection of methods for the CCAT data set. As with the MX-PO data set, the classifier that is mostly selected is *zarbi*, used for 68% of the authors, *naïve*, *neural* and *lssvm* come next, whereas *logitboost* and *rf* were not selected. The BER for linear classifiers is below the average BER for **FMS/1**, while the BER of non-linear methods is above the mean, giving evidence of the linearity of the problem. Most of the selected models included methods for preprocessing and feature selection. The BER of classifiers that used feature selection methods was higher than that of classifiers that were not used. The most used feature selection method was $pc - extract$, used for 19 models; other considered methods were *Ftest* (5), *Ttest* (5), *aucfs* (4) and *svcrfe* (3).

Table 5. Statistics on selection of methods when using PSMS for the CCAT collection

Classifiers						Feature Selection		Preprocessing	
zarbi	**naïve**	**neural**	**svc**	**kridge**	**lssvm**	**With**	**Without**	**With**	**Without**
Frequency of selection									
68%	10%	10%	2%	2%	8%	76%	24%	88%	12%
BER									
14.22	9.88	23.42	3.82	44.01	33.51	14.14	22.28	15.64	16.50

Figure 1 shows the per-author F_1−measure, the best result obtained was for the author *'Karl-Penhaul'* ($F_1 = 96.91\%$), which considered the three preprocessing methods, and *Ftest* for feature selection together with a naïve classifier. The classifier was built on 104 out of the 3,400 features (i.e., words) available, this means that about 100 words are enough for distinguishing this author; interestingly, 35 out of the 104 words selected as relevant were not used in any document of this author, the relevant words *'state'* and *'also'* were used in 41 out of 50 documents written by *'Karl-Penhaul'*.

On the other hand, the worst result was obtained for *'Peter-Humphrey'* ($F_1 = 14.81\%$), which used *normalize* for preprocessing and an *lssvm* classifier. When we used the *zarbi* classifier with the **FMS/0** setting, the classifier selected for this author obtained $F_1 = 45.71\%$, such classifier used the three preprocessing methods and *Zfilter* for selecting the top 234 more relevant features. This represents an improvement of over 30% in terms of F_1 measure, and an important improvement in terms of processing time, also, the result suggest the author *'Peter-Humphrey'* can be better modeled with a linear classifier.

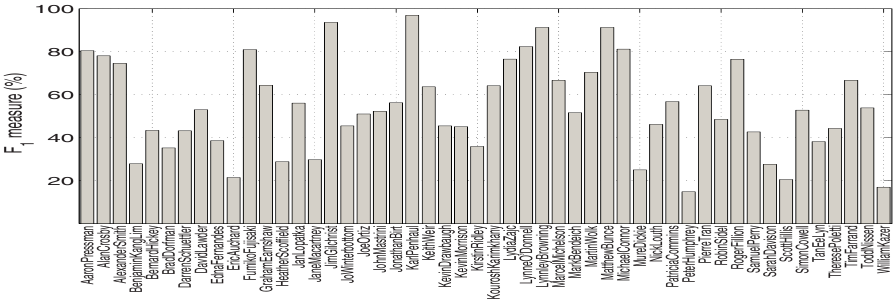


Fig. 1. F_1 measure of different classifiers in CCAT collection

5 Conclusions

We have described the application of particle swarm model selection (PSMS) to the problem of authorship verification. The proposed approach allows us to model each author independently, developing *ad-hoc* models for each author. This is an advantage over previous work that has considered a same learning algorithm for all of the authors. PSMS also selects methods for preprocessing and feature selection, facilitating the design and implementation processes to users. Experimental results show that the proposed technique can obtain reliable models that perform better than those in which the same learning algorithm is used for all of the authors. Results are satisfactory, despite we have used the simplest set of features one may try (i.e., the bag-of-words representation); better results are expected by using more descriptive features. PSMS can also be helpful for analyzing what features are more important for building classifiers for certain authors, which allows us to gain insight into the writing style of authors. Future work includes extending the use of PSMS for the task of authorship attribution and analyzing the writing style of authors by using models selected with PSMS.

Acknowledgements. We thank E. Stamatatos for making available the CCAT data and reviewers for their comments that have helped us to improve this paper. This work was supported by CONACyT under Project Grant CB-2007-01-83459.

References

1. Argamon, S., Marin, S., Stein, S.: Style mining of electronic messages for multiple authorship discrimination. In: Proc. of SIGKDD 2003, pp. 475–480 (2003)
2. Coyotl-Morales, R.M., Villaseñor-Pineda, L., Montes-y-Gómez, M., Rosso, P.: Authorship attribution using word sequences. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 844–853. Springer, Heidelberg (2008)
3. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence. Wiley, Chichester (2006)
4. Escalante, H.J., Montes, M., Sucar, E.: Particle swarm model selection. Journal of Machine Learning Research 10, 405–440 (2009)
5. Gorissen, D., Tommasi, L., Croon, J., Dhaene, T.: Automatic model type selection with heterogeneous evolution. In: Proc. of WCCI 2008, pp. 989–996 (2008)
6. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Analysis of the IJCNN 2007 ALvsPK challenge. Neural Networks 21(2–3), 544–550 (2008)
7. Van Halteren, H.: Linguistic profiling for author recognition and verification. In: Proc. of ACL 2004, pp. 199–206 (2004)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
9. Houvardas, J., Stamatatos, E.: N-gram feature selection for author identification. In: Euzenat, J., Domingue, J. (eds.) AIMSA 2006. LNCS (LNAI), vol. 4183, pp. 77–86. Springer, Heidelberg (2006)
10. Koppel, M., Schler, J.: Authorship verification as a one-class classification problem. In: Proc. of ICML 2004, p. 62 (2004)
11. Luyckx, K., Daelemans, W.: Authorship attribution and verification with many authors and limited data. In: Proc. of COLING 2008, pp. 513–520 (2008)
12. Momma, M., Bennett, K.: A pattern search method for model selection of support vector regression. In: Proc. of SIAM-CDM (2002)
13. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2006)

Image Characterization from Statistical Reduction of Local Patterns

Philippe Guermeur and Antoine Manzanera

ENSTA - Elec. and Comp. Sc. lab
32 Bd Victor, 75015 Paris, France
philippe.guermeur@ensta.fr
http://www.ensta.fr/~guermeur

Abstract. This paper tackles the image characterization problem from a statistical analysis of local patterns in one or several images. The induced image characteristics are not defined a priori, but depends on the content of the images to process. These characteristics are also simple image descriptors and thus considering an histogram of these elementary descriptors enables to apply "bags of words" techniques. Relevance of the approach is assessed when dealing with the image recognition problem in a robot application framework.

Keywords: Image recognition, Vector quantization, Histogram comparison.

1 Introduction

Local image description techniques usually relate to interest point detection methods. Many image processing methods define the notion of interest point from using a theoretical model of gray level variations on a local image neighbourhood [1], [2], [3]. More recent approaches combine interest points to image descriptors. A typical approach is the SIFT method which computes histograms from gradient orientations near interest points locations. [4] gives an overview of these various approaches which have often been used in "bags of words" methods for image indexing applications or robot navigation tasks such as Simultaneous Localization and Mapping of the Environment (SLAM). All of these techniques define the notion of interest point from designing an a priori model of what is a corner or some other interesting geometric element and so elaborating a model of what is the ideal gray level variation in the neighbourhood of the corresponding image points.

Others approaches aim at analysing images through statistical studies concerning the appearance of local image configurations (patterns). Thanks to a coding phase the number of patterns can be decreased and the Zipf law can be applied to model their distribution in the images [5]. This model enables to characterize texture complexity but gives no piece of spatial information as for the interest area location. This difficulty can be got round by previously partitioning the image into small regions before carrying out any statistical analysis. This is

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 571–578, 2009.

the way [6] characterizes every region with a 58-value local binary pattern (LPB) histogram and then concatenates the various image histograms to compose a single vector which is expected to be peculiar to the image being processed. This method has been efficiently experimented in facial expression recognition. In [7] a codebook is built from quantizing gray level image neighbourhoods and used within the context of texture recognition.

Our study is concerned primarily with automatic classification of basic image patterns, without any prior knowledge regarding their configuration. These patterns are local image neighbourhoods, we propose to transform before processing, in order to give the approach the desired invariance properties. How many classes are obtained, depends on a similarity threshold given as input to the algorithm. Next, we propose to characterize the images from the statistical properties of the detected patterns, through "bags of words" techniques. A pattern codebook is built from a subset of images (learning step) and then applied to generate a feature histogram for every image (analysis step). Section 2 gives details about this extraction of characteristics. The various histograms are stored in a database for being used as feature vectors later. Image recognition tasks can be made by comparing the new histogram with the registered individual histograms (section 3) and selecting the best match as the recognition result. Section 4 evaluates the effectiveness of this method in a piece recognition framework with several comparison functions.

2 Characteristics Extraction

This section deals with the technique of codebook generation, including a K-means classification method.

2.1 Algorithm

We consider very basic image structures composed of image neighbourhoods. For each neighbourhood, we assume to be of size k (let us note $k = n \times n$), we pick up the pixels intensities to compose vectors in a k-dimensional Euclidean space in case of a gray level image, or in a mk-dimensional space in case of a m-component image (e.g. colour image).

An incremental (K-means modified) clustering method is applied to construct the codebook (learning step). Then, during the analysis step every new vector is identified to a codeword. The codebook construction is based on three steps:

```
Step 1. Begin with an initial codebook.
Step 2. For each neighbourhood, find the nearest neighbour in the
codebook.
Step 3. If the distance to this codeword is less than a given
radius denoted dmax afterwards, compute the new centroid and
replace the codeword with it. Otherwise add the new word to the
codebook.
```

2.2 Preliminary Image Transformation

The clustering algorithm can be applied directly on the original image, but in order to give the algorithm some convenient invariance properties a preliminary image transformation should be useful. The overall process is illustrated in Fig. 1 where the pre-processing step is represented with the spatio-temporal function Ψ.

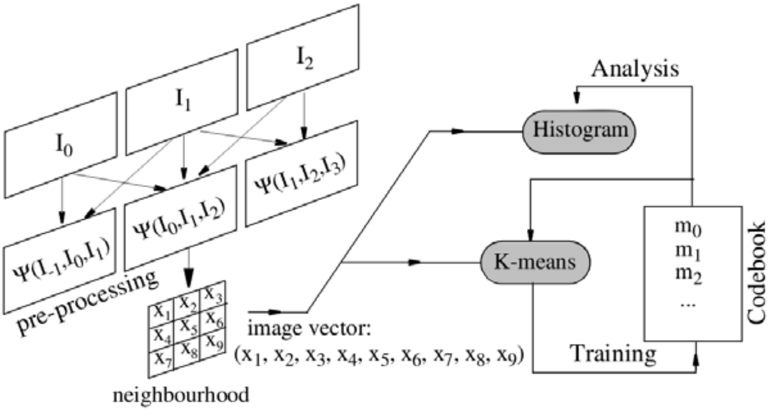


Fig. 1. Overall principle of the algorithm (with $k = 9$)

As we are more particularly concerned with invariance to lightning change, we propose an implementation in which the output of the pre-processing step represents the image gradient arguments. We suggest to code the resulting image data with 16 values, which means a $\pi/4$ quantization step. When considering colour images, only the argument of the gradient which have the maximal magnitude (computed as in [11]) is kept, if this magnitude is above a given threshold.

2.3 Distance Function

Whether being concerned with the training step or with the analysis step, the vectors extracted from an image have to be compared to those gathered in the dictionary. The metric to use depends on whether or not the image has been pre-processed.

Referring to an original image, we propose this metric to be represented by the Euclidian distance between the two vectors after they have been normalized. Let us assume that x and y are the two normalized vectors to compare, the distance between these vectors is given by:

$$E = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \qquad (1)$$

As this metric is representative of the angle and does not consider the vector modulus, it is supposed to be invariant to illumination changes. The algorithm elaborated with this metric will be denoted algorithm 1 afterwards.

The alternative technique we propose to apply consists in pre-processing the original image in order to make each of the vector components invariant to illumination changes. So, we propose that this step comes down to replace each pixel by a representation of the argument of the gradient. In order to keep only the significant values, only the pixels whose gradient is above a given threshold are considered.

A quantization step of $\pi/4$ leads us to get a 16-value code that have to be completed with an additional value to code the non-significant data. Regardless to the original image the resulting data are reduced and a simple distance can be elaborated to compare two vectors. We propose to use the following one, based on the $\mathcal{L}_1$ norm:

$$E = \sum_{i=1}^{N} |\alpha_i - \beta_i| \tag{2}$$

where α_i and β_i are respectively the ith component of the two vectors to compare. The algorithm elaborated with this metric will be denoted algorithm 2 afterwards.

3 Histogram Comparison

The analysis step consists in searching for every image neighbourhood, the nearest vector in the codebook. The various results enable either to quantize the original image or to implement an efficient histogram algorithm. At the end of this step, any image is supposed to be characterized by a single histogram and so comparing the obtained histograms between them should indicate how much the corresponding images are similar. This section addresses this problem of histogram comparison through a brief presentation of a few possible comparison functions drawn from statistics, signal processing and geometry. Table 1 gives expressions for such convenient functions, most of them found in the literature, by dividing them into two groups according to their origin. In this table, we assume the vectors X and Y represent the two histograms to be compared, and the vector A is an average histogram as far as such a piece of information is available.

4 Application to Place Recognition

The experiments have been conducted on the INDECS database [9][10], which contains pictures of five different rooms acquired at different times of the day under different viewpoints and locations. The system is trained for all the pictures acquired under a given illumination condition and tested under the other illumination conditions for the remaining pictures. For a given picture, the aim is to recognize the room in which it has been acquired.

Table 1. Distance functions considered for histogram comparison

min/max based functions	χ^2 and Bhattacharyya [8] based functions
$\bigcap(X,Y) = \sum_i \min(X_i,Y_i)$	$\chi^2(X,Y) = \sum_i \frac{(X_i - Y_i)^2}{Y_i}$
$\bigcap_{av}(X,Y) = \sum_i \frac{1}{A_i} \min(X_i,Y_i)$	$\chi^2_m(X,Y) = \sum_i \frac{(X_i - Y_i)^2}{(X_i + Y_i)}$
$\bigcap_{yav}(X,Y) = \sum_i \frac{Y_i}{A_i} \min(X_i,Y_i)$	$\chi^2_{av}(X,Y) = \sum_i \frac{(X_i - Y_i)^2}{A_i(X_i + Y_i)}$
$\Psi(X,Y) = \sum_i \frac{\min(X_i,Y_i)}{\max(X_i,Y_i)}$	$Bha(X,Y) = \frac{\sum_i \sqrt{X_i Y_i}}{\sqrt{\sum_i X_i}\sqrt{\sum_i Y_i}}$
$\Psi_{av}(X,Y) = \sum_i \frac{\min(X_i,Y_i)}{A_i \max(X_i,Y_i)}$	$Bha_{av}(X,Y) = \frac{\sum_i \frac{\sqrt{X_i Y_i}}{A_i}}{\sqrt{\sum_i X_i}\sqrt{\sum_i Y_i}}$

The first tests have been done with our two algorithms in order to evaluate the different distance functions used for histogram comparison. They consist to evaluate the classification rate (that is the successful room recognition rate) for every image of the database, by giving each image the same weight. For better comparison with the results obtained in [9] (where the pictures containing less than 5 interest points were rejected) we choose to discard the images whose histogram contains no bin. This concerns about 1% of the total number of images. The results are plotted on figure 2 and figure 3. They indicate that the best results are obtained with Bhattacharyya metric when we consider algorithm 1 and the χ^2_m metric when we consider algorithm 2. Using this last metric, the classification rate of algorithm 2 are around 80% and the best rate is 82.42% with $dmax = 16$. The performances of algorithm 1 are weaker, but the overall performance increases when *dmax* decreases. However, the evaluation has not been done for *dmax* inferior to 0.35 as for this value the algorithm 1 has a great number of clusters (6 319 clusters) and consequently becomes too slow. Figures 4(a) and 4(b) illustrate the swift variation of the number of clusters when *dmax* decreases, for both algorithms.

The classification rate is then calculated separately for each room and according to [9] a single measure of performance is computed by averaging all the results with equal weights. These results are plotted on figure 5 and figure 6, with regard to the results obtained in [9] and [10] which use a local feature detector constructed from the combination of the Harris-Laplace detector and the SIFT descriptor. These figures clearly indicate that the performance measure obtained with our algorithm is the best, regardless of which training set is chosen.

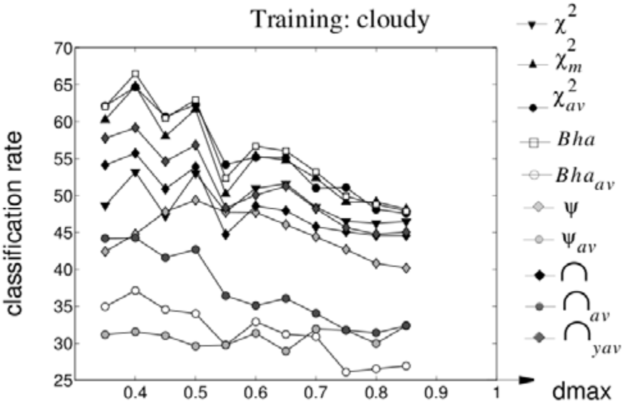


Fig. 2. Global place classification rate using algorithm 1

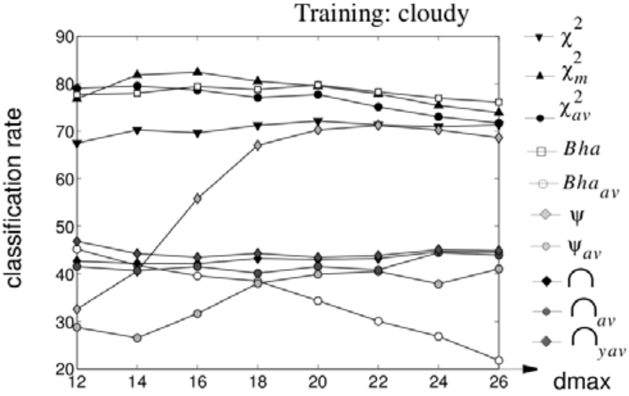
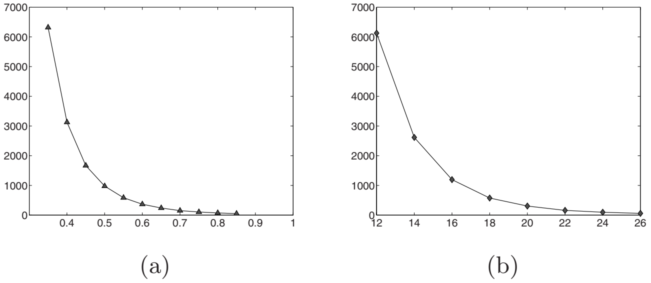


Fig. 3. Global place classification rate using algorithm 2



(a) (b)

Fig. 4. Number of clusters with regard to the radius *dmax*, using (a) Algorithm 1, or (b) Algorithm 2

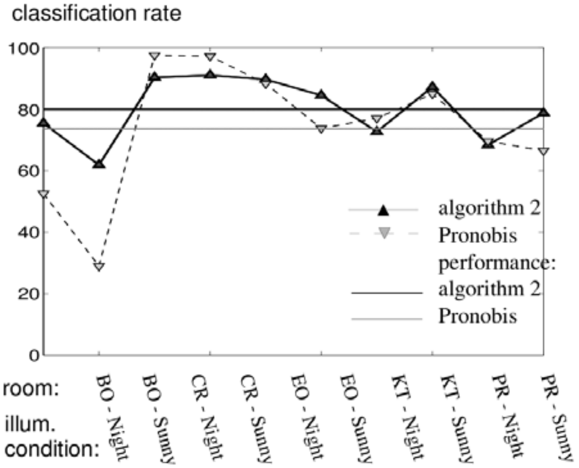
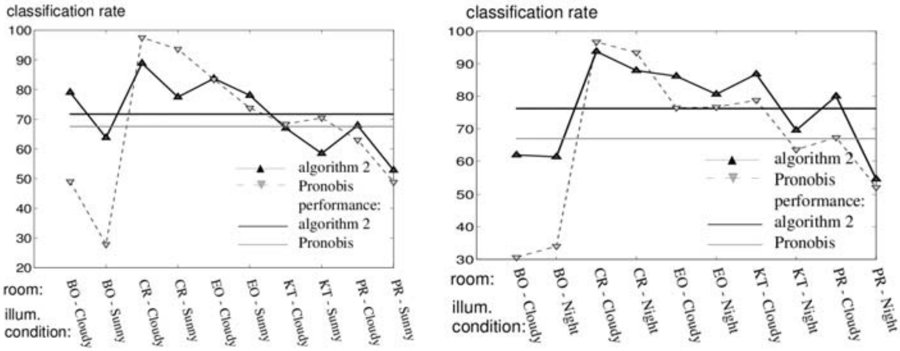


Fig. 5. Classification rates and performance measure after training the algorithm with "cloudy" illumination condition



(a) (b)

Fig. 6. Classification rates and performance measure after training the algorithm with: (a) "nighty" or (b) "sunny" illumination condition

5 Conclusion

This study has tackled the image characterization problem by proposing a new approach based on a statistical analysis. Though this approach is very simple, it has been proven to be efficient and it has been empirically validated in the robotic framework of place recognition. Studies are in progress in order to improve the classification rates (splitting of the image into small regions, higher order statistic reduction) or to lead to real-time implementations of the proposed method.

Acknowledgments

The authors thank Maurice Diamantini for his contribution in programming with Ruby. This work was sponsored by the DGA. The support is gratefully acknowledged.

References

1. Wang, H., Brady, M.: Real-time corner detection algorithm for motion estimation. Image and Vision Computing 13(9), 695–703 (1995)
2. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceeding of the 4th Alvey Vision Conference, pp. 147–151 (1988)
3. Trajkovic, M., Hedley, M.: Fast corner detection. Image and Vision Computing 16(2), 75–87 (1998)
4. Mikolajczyk, K., Schmid, C., et al.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
5. Caron, Y., Charpentier, H., Makris, P., Vincent, N., et al.: Une mesure de complexité pour la détection de la zone d'intérêt d'une image. In: DGCI 2003 (2003)
6. Feng, X., Pietikäinen, M., Hadid, A., et al.: Facial Expression Recognition Based on Local Binary Pattern. Pattern Recognition and Image Analysis 17(4), 592–598 (2007)
7. Varma, M., Zisserman, A.: A Statistical Approach to Material Classification Using Image Patch Exemplars. To be published in IEEE PAMI
8. Thacker, N.A., Aherne, F.J., Rockett, P.I.: The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data, Kybernetika, Prague, June 9-11 (1997)
9. Pronobis, A.: Indoor Place Recognition Using Support Vector Machines, Master's Thesis in Computer Science (2005)
10. Pronobis, A., Caputo, B., Jensfelt, P., Christensen, H.I.: IA Discriminative Approach to Robust Place Recognition. In: ICIRS, Beijing, China (2006)
11. Zenzo, S.D.: A note on the Gradient of a multi-image. In: CVGIP, vol. 33, pp. 116–125 (1986)

Semi-supervised Robust Alternating AdaBoost*

Héctor Allende-Cid[1], Jorge Mendoza[2], Héctor Allende[1,2], and Enrique Canessa[2]

[1] Universidad Técnica Federico Santa María,
Dept. de Informática, Valparaíso-Chile
vector@inf.utfsm.cl

[2] Universidad Adolfo Ibáñez,
Facultad de Ingenieria y Ciencias, Viña del Mar-Chile
jorge.mendoza2003@alumnos.uai.cl, hallende@uai.cl, ecanessa@uai.cl

Abstract. Semi-Supervised Learning is one of the most popular and emerging issues in Machine Learning. Since it is very costly to label large amounts of data, it is useful to use data sets without labels. For doing that, normally we uses Semi-Supervised Learning to improve the performance or efficiency of the classification algorithms.

This paper intends to use the techniques of Semi-Supervised Learning to boost the performance of the Robust Alternating AdaBoost algorithm.

We introduce the algorithm RADA+ and compare it with RADA, reporting the performance results using synthetic and real data sets, the latter obtained from a benchmark site.

Keywords: Semi-Supervised Learning, Expectation Maximization, Machine ensembles, Robust Alternating AdaBoost.

1 Introduction

In supervised learning, classification tasks require training data with a class label. However, there are many real problems where the existence of labeled data is scarce and unlabeled data is abundant, either due to its cost or because it is difficult to obtain it (i.e. classification of text and web pages, processing medical imaging, diagnosis of industrial processes, speech recognition, protein structures, etc.). For this reason, it is necessary to build classifiers that work with a small amount of labeled data and a large amount of unlabeled data so they can learn from both. The main idea behind the algorithm RADA [1] (acronym for Robust Alternating AdaBoost) is to alternate the use of classical and inverse AdaBoost in order to lessen the impact of data outliers in the final classification.

In this paper we propose a generalization of the algorithm RADA for use in Semi-Supervised learning problems. Basically, the aim is to make use of the robust properties of the algorithm and extend it to take advantage of unlabeled

* This work was supported in part by Research Grant Fondecyt 1070220, and DGIP-UTFSM Grant.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 579–586, 2009.

data to train the weak classifiers. In Section 2 we briefly introduce the algorithm RADA. Section 3 presents the analysis of the generalization of RADA to Semi-Supervised classification. In Section 4 we present the proposed algorithm RADA+. Experimental results are presented with both synthetic data and real data in Section 5. The last section is devoted to concluding remarks.

2 Robust Alternating AdaBoost

RADA is an acronym for Robust Alternating AdaBoost. As its name suggests, this algorithm combines the power of classical and inverse AdaBoost to reduce the impact of data outliers in training samples. The RADA algorithm bounds the influence of the outliers to the empirical distribution, it detects and diminishes the empirical probability of "bad" examples, and it performs a more accurate classification under contaminated data.

RADA computes the relative weight of each instance in the training set using a different equation. Originally AdaBoost obtains the relative weight as follows:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{1}$$

RADA uses a robustified equation of α_t for smoothing the impact of an outlier data:

$$\alpha_t = \begin{cases} \frac{1}{2}\sqrt[r]{\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)} + \alpha_\gamma & \epsilon_t < \gamma \\ \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) & \epsilon_t \geq \gamma \end{cases} \tag{2}$$

where $\alpha_\gamma = \frac{1}{2}\ln\left(\frac{1-\gamma}{\gamma}\right) - \frac{1}{2}\sqrt[r]{\ln\left(\frac{1-\gamma}{\gamma}\right)}$ is a constant needed so that equation (2) is continuous.

Applying the previous equation will prevent the empiric distribution from growing considerably in one step for any sample. However, the empirical distribution is updated at each stage, and after a few iterations the probability weight of the samples that were misclassified repeatedly, will have bigger values compared to other samples. To solve this problem, Allende-Cid et al. [1] introduces two new variables to the algorithm: an inverting variable $\beta(i)$ and an age variable $age(i)$ for each example $i = 1 \ldots n$.

When the variable $\beta(i)$ value is 1, the algorithm behaves as the classical AdaBoost, i.e., the empirical distribution increases when a sample is misclassified and decreases the value otherwise. If the value of $\beta(i) = -1$, then the algorithm behaves like the Inverse AdaBoost, i.e., decreases the empirical distribution when a sample is misclassified and increases it otherwise. The variable $age(i)$ counts the number of times a sample i has been misclassified, if the number exceeds a threshold τ then the value of $\beta(i)$ is changed to -1 (originally the value of $\beta(i)$ for all samples is 1).

3 RADA Semi-supervised Generalization

The algorithm RADA compares the actual values of the class versus the value estimated by the ensemble to update the sampling distribution. Thus those instances that have been difficult to classify, i.e., the classification of the weak classifier differs from the actual class, will have a bigger probability to be selected in the training set on the next iteration. One of the main problems that arise from the method presented is that an unlabeled data has no "real" class to compare it with.

RADA algorithm defines the margin of an instance obtained in the i-th iteration by the equation

$$\alpha_T \beta(i) y_i h_T(x_i) \tag{3}$$

The problem, therefore, lies in defining the margin for an unlabeled data. To solve this problem, we take advantage of the cluster and manifold assumptions [5]. This seeks to improve the margin of classification (equivalent to minimize the error of the ensemble) through the selection of unlabeled data with a higher confidence rating, and assigns the class predicted by the current classifier.

To allow the same margin to be used for both labeled and unlabeled data, we use the same definition of *pseudo-class* as [4]. A subset of labeled data in addition to a group of unlabeled data and their pseudo-class, are used for training the weak classifier in the next iteration. This same strategy is used by algorithms such as ASSEMBLER [4], Self-Training [10] and Semisupervised MarginBoost [6].

First we find a mechanism to define the margin of an unlabeled data. For that we use the same function used in RADA, defining the base classifier as $h_T(x) : \mathbb{R} \rightarrow \{-1, 1\}$ where h_T is the T-th classifier in the ensemble. The set of training labeled data, L, is n-dimensional type of $x_1, x_2, \ldots x_n$ with their respective classes $\{-1, 1\}$. The classifier of the ensemble $H_T(x)$ is a linear combination of classifiers in step T

$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{4}$$

where α_t is the equivalent weight in the algorithm RADA.

Now when adding a unlabeled data set, U, a margin associated with these data must be defined (as in the case of labeled examples). However, there is no knowledge of the true value of the class of the data, so following [6,3,8,11] we define the margin for an unlabeled data x_i as

$$|h_T(x_i)| \tag{5}$$

since the above expression is an absolute value, one can apply a mathematical simplification to represent this term

$$y_i h_T(x_i) \tag{6}$$

This allows to generalize the concept for both labeled and unlabeled data. If x_i is a labeled data, then y_i is the known class, on the contrary, if x_i is a unlabeled data, then y_i represents the pseudo-class (as defined above).

Once the problems of the margin were solved, we used the framework Expectation-Maximization (EM) [9]. EM is a popular iterative algorithm for maximum likelihood estimation in problems with unlabeled data. It consists two steps: the Expectation step and the Maximization step. The first step consists in classifying the unlabeled data based on the current hypothesis. Then the Maximization step re-estimates the parameters based on all the data (labeled and unlabeled with a pseudo-class). This leads to the next iteration of the algorithm, and so on. It has been proved that EM converge to a local minimum when the model parameters stabilize [9].

4 Semi-supervised Robust Alternating AdaBoost

In this section we present the proposed algorithm Semi-Supervised Robust Alternating AdaBoost (RADA+). The main idea of this proposal is to add unlabeled data, after a certain number of training epochs j, to the training data set in order to enhance the overall performance of the algorithm.

The developed framework is the following: At first we will take the labeled data and will train a non-ensemble based supervised classifier with it (in this particular case we took the SVM algorithm and trained it with the labeled data). After a number of j training epochs we compare the result of the H_j classifier with the SVM algorithm. If the result obtained from both of the classifiers is the same, we add these data examples to the training data set, hopefully to enrich the training phase of the algorithm. From the $j+1$ iteration on we use an EM approach to update the classification of the unlabeled data with the strong classifier H_{j+1}. The fundamentals behind our approach is to prove the impact of the strong classifier H_j on the weak classifiers. For this reason we propose to add the unlabeled data to the training data set in an iteration where the strong classifier is robust enough so as not to affect the final classification.

The parameters are defined in the following way:

$$D_j(x_U) = \max_{x \in \mathcal{U}} D_j(x), \quad age(i) = 0, \quad \beta(i) = 1$$

$D_j(x_U)$ is the initial weight of the unlabeled data examples when they are added to the training data set. We chose the maximum because we think that it is important that these examples are chosen when the resampling is made.

Algorithm 1 shows our proposed Semi-Supervised Alternating AdaBoost algorithm.

5 Experimental Results

In this section we empirically show the performance of our Semi-Supervised RADA (**RADA+**) model proposal compared to the **RADA** algorithm, for both

Algorithm 1. RADA+ Algorithm

```
Training Data Set $\mathcal{L} = \{(x_1, y_1), \ldots, (x_{n_l}, y_{n_l})\}$ with $n_l$ elements, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$
Unlabeled Data Set $\mathcal{U} = \{x_{n_l+1}, \ldots, x_{n_l+n_u}\}$ with $n_u$ elements, where $x_i \in \mathcal{X}$
Choose: $\tau$ age threshold, $\gamma$ limit threshold, the robust parameter $r$ and $t = 0$.
Train the non-ensemble based classifier with the training data set. Then perform a classification of the unlabeled data $\mathcal{U}$ with the classifier and assign them the corresponding pseudo-class $y$.
$D_1(i) = \frac{1}{n_l}$, $\beta(i) = 1$ and assign the $age(i) = 0$ variable to each sample $(x_i, y_i)$, $i = 1, \ldots, n_{n_l}$
repeat
    Increment $t$ by one.
    Select a bootstrap sample $Z_t$ from $\mathcal{Z}$ with distribution $D_t$.
    Construct $h_t : \mathcal{X} \rightarrow \{-1, +1\}$ classifier using the bootstrapped sample $Z_t$ as the training set.
    Calculate the ensemble error in step $t$:
        $$\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i}^{n} D_t(i)$$
    Calculate $\alpha_t$ as in (2).
    if $t = j$ then
        Classify the unlabeled data $\mathcal{U}$ with $H_{t-1}$.
        if $H_{t-1}(x_u) = y(x_u)$ then
            Add $x_u$ to the training set $\mathcal{Z}$
            Set the distribution $D_j(x_u) = \max_{x \in \mathcal{Z}} D_j(x)$, $age(x_u) = 0$ and $\beta(x_u) = 1$
        end if
    end if
    if $t \geq j + 1$ then
        Classify the unlabeled data that entered in the iteration $j$ with $H_{t-1}$
        Update the pseudo-class
    end if
    Update distribution
        $$D_t : D_{t+1} = \frac{D_t(i)}{W_t} \times e^{(-\alpha_t \beta(i) y_i h_t(x_i))}$$
    where $W_t = \sum_i D_t(i)$
    Final hypothesis $H_t$ in iteration $t$ is given by:
        $$H_t = sign\left(\sum_{k=1}^{t} \alpha_k h_k(x)\right)$$
    Classify $\mathcal{Z} = (x_1, y_1), \ldots, (x_n, y_n)$ with $H_t$
    if Sample $(x_i, y_i)$ was correctly classified by $H_t$ (meaning that $H_t y_i > 0$) then
        $age(i) = 0$ y $\beta(i) = 1$
    else
        Increment $age(i)$ by one
        If $age(i) > \tau$ then $\beta(i) = -1$ and $age(i) = 0$
    end if
until Stopping criterion is met
Output: hypothesis $H_t(x)$
```

Synthetic and Real data sets; the latter was obtained from the UCI Machine Learning Repository [2].

The data of both synthetic and real data sets were separated in labeled, unlabeled and test sets. The results reported for each model correspond to the mean value of the computed metrics, over 20 runs, using the same data sets.

For the synthetic data we used the following proportion labeled/unabeled data: 1%, 5% and 10%. For the real data sets, we used the following proportion labeled/unlabeled data: 5%, 10% and 20%. The difference lies on the data sets sizes, for the synthetic sets the total amount of data analyzed was 15000 instances, instead for the real data sets the amount of total data was approximately 5000 instances. The classifier used in the algorithms is the Bayesian Classifier (QDA) (see [7]). The non-ensemble based clasifier used to determine whether to add unlabeled data to the test set or not, was a Soft-margin Support Vector Machine with a Sequential Minimal Optimization method to find the separating hyperplane.

For the synthetic experiment we created a synthetic data set $\{(x_i, y_i)\}_{i=1}^{n}$, as an independent sample obtained from a mixture of gaussian distributions labeled with the class $\{-1, 1\}$. For more information on the details of the synthetic data sets, please refer to [1].

Table 1 shows the summary results of the performance evaluation on the synthetic data of the RADA and RADA+ algorithms, with 1%, 5% and 10% labeled data. As we can observe in the *Test Error* column, RADA and RADA+ have very similar behavior specially for the presence of a low percentage of labeled data. However, this radically changes when the amount of labeled data increases, i.e. 10%. Nevertheless RADA+ obtained good results in the training set, mainly because of the EM framework.

Table 1. Summary results of the performance evaluation of the RADA and RADA+ algorithms with 5% and 10% outliers

Labeled	Algorithm	Outliers	T	Train Error	Train Min.	Test Error	Test Min.
1%	RADA	5%	33.83	23.87 ± 10.44	15.19	**25.72 ± 9.48**	17.45
	RADA+	5%	15.8	**7.56 ± 11.52**	0.42	25.88 ± 4.52	20.14
	RADA	10%	32.1	26.71 ± 7.79	16.36	26.21 ± 7.12	16.81
	RADA+	10%	13.7	**8.04 ± 12.28**	0.50	**25.69 ± 4.05**	18.63
5%	RADA	5%	25.7	25.58 ± 1.55	24.16	25.56 ± 1.47	24.11
	RADA+	5%	20.2	**9.06 ± 10.49**	2.36	**24.52 ± 1.72**	22.25
	RADA	10%	15.1	23.36 ± 0.84	22.65	**23.87 ± 0.83**	23.14
	RADA+	10%	14.6	**8.60 ± 9.81**	2.28	25.07 ± 1.83	22.31
10%	RADA	5%	14.1	25.46 ± 1.16	24.08	**25.42 ± 1.14**	24.07
	RADA+	5%	2.6	**11.65 ± 8.20**	3.52	39.84 ± 11.75	24.37
	RADA	10%	16.7	23.25 ± 0.95	22.26	**23.36 ± 0.87**	22.47
	RADA+	10%	6.7	**11.24 ± 7.81**	3.63	36.03 ± 11.01	22.81

Table 2. Summary results of the performance evaluation of the RADA and RADA+ algorithms with real data sets

Data sets	% Labeled	Algorithm	T	Train Error	Train Min.	Test Error	Test Min.
Page Blocks	5%	RADA	30.9	7.75 ± 6.22	0.73	9.55 ± 5.86	2.89
		RADA+	16.2	**0.67 ± 1.04**	0.07	**3.24 ± 0.38**	2.91
	10%	RADA	22.6	8.50 ± 8.52	1.69	9.72 ± 8.41	3.03
		RADA+	22.4	**0.90 ± 1.17**	0.22	**3.48 ± 0.53**	3.08
	20%	RADA	20.5	8.57 ± 1.96	7.11	7.90 ± 1.83	6.61
		RADA+	29.3	**3.46 ± 4.35**	0.95	**4.93 ± 3.14**	2.96
Wave Forms	5%	RADA	9.2	**0.35 ± 1.34**	0.00	11.29 ± 0.44	10.23
		RADA+	24.5	0.40 ± 1.55	0.00	**10.73 ± 0.55**	9.99
	10%	RADA	2.5	**0.99 ± 2.17**	0.13	12.77 ± 0.79	10.53
		RADA+	19.3	1.00 ± 2.15	0.02	**11.21 ± 0.62**	10.30
	20%	RADA	5.4	3.39 ± 2.73	1.42	11.09 ± 0.43	10.05
		RADA+	30.0	**2.69 ± 3.09**	0.71	**10.26 ± 0.49**	9.62

We tested two real data sets: Page Blocks and Wave Forms. In these data sets we changed the number of classes, mainly because both data sets had more than two. Table 2 shows the summary results of the performance evaluation on these real data sets of the RADA and RADA+ algorithms.

We must note that as the training information decreases, the performance gap between the proposed algorithm RADA+ and RADA becomes larger. Note that the difference in the training error is quite noticeable. This is due to the use of the framework EM in the algorithm, specially when the labeled data is scarce, which is the same result obtained for the synthetic data sets. In the T column, we observe a different behavior regarding the results obtained for the synthetic data sets. The number of iterations is always for the RADA+ algorithm than for RADA, however the minimum test error is lower, wish indicates that RADA+ reaches a smaller error.

6 Concluding Remarks

The results were mixed, mainly because of the difference in the data sets used in experiments. In the real data sets, RADA+ outperforms RADA in both of the data sets, however the results obtained in the Page Blocks experiments were better than the ones obtained in Wave Forms. In the synthetic data set the performance of RADA+ was only slightly better than the one obtained with RADA, but still there was an improvement.

It is important to analyze the effect of the algorithm Support Vector Machine (SVM) in the proposal. The SVM is an algorithm widely used in classification tasks, unfortunately it has a bad performance in the presence of data outliers. Thus, the use of SVM in this proposal is beneficial for noiseless data, but for noisy data it is rather harmful. This behavior is observed in the synthetic data sets, where the results obtained where not as good as the ones obtained in the real data sets. This leaves open the opportunity to explore the use of different supervised algorithms with the proposed RADA+. It is also likely that the semi-supervised learning paradigm suffers from outliers. Since it tries to use distributional information from the unlabeled data, if the data contains outliers that discovered distributional information might be misleading. Further studies are needed to prove these conclusions.

This paper does not intend to corroborate the robustness properties of the algorithm RADA, but rather use the concepts of Semi-Supervised Learning to improve performance of the algorithm with large amounts of unlabeled data. The experimental results proved that the performance of the RADA+ algorithm is better than the one for RADA under those conditions.

References

1. Allende-Cid, H., Salas, R., Allende, H., Ñanculef, R.: Robust Alternating AdaBoost. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 427–436. Springer, Heidelberg (2007)
2. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
3. Bennett, K., Demiriz, A.: Semi-Supervised Support Vector Machines. Advances in Neural Information Processing Systems, 368–374 (1999)
4. Bennett, K.P., Demiriz, A., Maclin, R.: Exploiting unlabeled data in ensemble methods. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 289–296 (2002)
5. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. MIT Press, Cambridge (2006)
6. Alche-Buc, F.d., Grandvalet, Y., Ambroise, C.: Semi-Supervised MarginBoost. Advances in Neural Information Processing Systems 1, 553–560 (2002)
7. Duda, R., Hart, P., Stork, D.: Pattern classification. Wiley-Interscience, Hoboken (2000)
8. Grandvalet, Y., d'Alché-Buc, F., Ambroise, C.: Boosting Mixture Models for Semi-supervised Learning. In: Dorffner, G., Bischof, H., Hornik, K. (eds.) ICANN 2001. LNCS, vol. 2130, pp. 41–48. Springer, Heidelberg (2001)
9. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal processing magazine 13(6), 47–60 (1996)
10. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: Seventh IEEE Workshop on Applications of Computer Vision, vol. 1, pp. 29–36 (2005)
11. Vapnik, V.N.: Statistical learning theory. John Wiley & Sons, Chichester (1998)

Robust Radio Broadcast Monitoring Using a Multi-Band Spectral Entropy Signature

Antonio Camarena-Ibarrola[1], Edgar Chávez[1,2], and Eric Sadit Tellez[1]

[1] Universidad Michoacana
[2] CICESE

Abstract. Monitoring media broadcast content has deserved a lot of attention lately from both academy and industry due to the technical challenge involved and its economic importance (e.g. in advertising). The problem pose a unique challenge from the pattern recognition point of view because a very high recognition rate is needed under non ideal conditions. The problem consist in comparing a small audio sequence (the commercial ad) with a large audio stream (the broadcast) searching for matches.

In this paper we present a solution with the *Multi-Band Spectral Entropy Signature* (MBSES) which is very robust to degradations commonly found on amplitude modulated (AM) radio. Using the MBSES we obtained perfect recall (all audio ads occurrences were accurately found with no false positives) in 95 hours of audio from five different am radio broadcasts. Our system is able to scan one hour of audio in 40 seconds if the audio is already fingerprinted (e.g. with a separated slave computer), and it totaled five minutes per hour including the fingerprint extraction using a single core off the shelf desktop computer with no parallelization.

1 Introduction

Monitoring content in audio broadcast consists in tagging every segment of the audio stream with metadata establishing the identity of a particular song, advertising, or any other piece of audio corresponding to feature programming. This tagging is an important part of the broadcasting and advertising businesses, all the business partners may use a third party certification of the content for billing purposes. Practical examples of application of this tagging include remote monitoring of audio marketing campaigns, evaluating the hit parade, and recently (in Mexico at least) monitoring announcements from political parties during election processes.

There are several alternatives to audio stream tagging or media monitoring, current solutions are ranged from low tech (e.g human listeners), to digital content tagging, watermarking and audio fingerprinting. In this paper we are interested in automatic techniques, where the audio stream can be analyzed and tagged without human intervention. There are several commercial turnkey solutions reporting about 97% precision with a very small number of false positives, the most renowned is *Audible Magic* `http://www.audiblemagic.com/`

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 587–594, 2009.

with massive databases of ads, songs and feature content. The core of the automated techniques is the extraction of an audio fingerprint, which is a succinct and faithful representation of the audio stream, in both the audio stream and the content to be found in the broadcast. This change of domain serve two purposes, on the one hand it is faster to compare the succinct representation. On the other hand, since only significant features of the signal are retained very high accuracy can be obtained in the comparison. In this paper we present a tagging technique for automatic broadcast monitoring based on the MBSES. Our technique has perfect recall and is very fast, scoring from 12 to 40 times faster than real time broadcasting in a single-core standard computer with no parallelization. As described in the experimental part we were able to improve the recognition rate of trained human operators working on a broadcast monitoring firm.

2 Related Work

It is a fact that most audio sources can be tagged prior to the broadcasting, specially with the advent of digital radio. Even in the case of analog audio broadcasting it is possible to embed digital data in the audio without audible distortion and persistent to degradations in the transmission. This technique, called *audio watermarking*, is suitable for applications where the broadcast station agree to modify the analog content, and needs a receiver capable of decoding the embedded data on the end point. This type of solutions are described in [1] and [2]. Usually they are sold as turnkey systems with both the transmitter and the receiver included. Watermarking is not suitable for doing audio mining or searching in large audio logs since in most of them (if not all), audio was not recorded with any embedded data.

A more general solution to Radio Broadcast Monitoring consist in making a succinct and faithful representation of the audio, specific enough to distinguish between different audio sequences and general enough to allow the identification of degraded samples. Common degradations are white/colored noise adding, equalization and re-recording. This technique is called *audio fingerprinting* and has been studied in a large number of scientific papers and due to its flexibility it has been the first choice mechanism for audio tagging. When small excerpts of audio are used to identify larger pieces of the stream an additional artifact is introduced to the process, the time shifting effect. This is due to the discrete audio window being represented, and the failure to match the start of the audio window in both the excerpt and the stream. Audio fingerprinting must be resilient to all the above distortions without loosing specificity. Several features have been used for audio-fingerprinting purposes, among them, the Mel-frequency Cepstral coefficients (MFCC) [3], [4]; the *Spectral Flatness Measure* (SFM) [5]; *tonality* [6] and *chroma values* [7], most of them are analyzed in depth in [8]. Recently in [9,10] the use of entropy as the sole feature for audio fingerprinting proved to be much more robust to severe degradations outperforming previous approaches. This technique is the *Multi-Band Spectral Entropy Signature* or MBSES described in some detail in this paper.

Once the fingerprint is obtained, it is not very difficult to build on this first piece a complete system for broadcast monitoring. Such a complete system is discussed in [11] using a fingerprint. In Oliveira's work [11] the relevant feature was the energy of the signal contained in both the time and the frequency domains. The authors reported a correct recognition rate of 95.4% with 1% of false positives. Another good example of a system for broadcast monitoring with excellent results is [12] where the relevant feature chosen was the *spectral flatness* which is also the feature used in the MPEG-7 wrapper (see [13] for details) for describing audio content.

Due to the economic importance of media monitoring (up to 5% of the total advertising budget is devoted to monitoring services) several companies have proprietary, closed technology for broadcast monitoring. In this case we can only compare with the performance figures publicly reported in white papers.

We selected MBSES to build our system due to its anticipated robustness. Using this fingerprint we were able to achieve perfect recall and no false positives in very low quality audio recordings just by tuning the time resolution. This results outperform the reported precision of both academic and industrial systems. Audio tagging, particularly using a robust fingerprint such as the one described in this paper, is a world class example of a successful pattern recognition technique. Several lessons can be extrapolated from this exercise.

The rest of this paper is organized as follows, first we explain how the MBSES of an audio signal is determined, then we describe the implemented system in detail, a description of the experiments performed to test our system follows, and finally some conclusions and future work directions are discussed in the last section.

3 Broadcast Monitoring with MBSES

The final product of a monitoring service is a tagged audio log of the broadcast. Assuming the role of the broadcast monitoring company, a particular client request counting a particular ad in a given number of radio stations. The search is for some common failures in the broadcasting of audio ads, namely the absence of the ad, airing it at a time different from the one paid (time slots have different prices depending of the time of the day, and the day itself) and airing only a fraction of the audio ad. Lack of synchronization between airing and marketing campaigns may lead to large loses, for example when a special offer that lasts one day only and the ads were aired the day after the special offer has expired. The only legal bonding for auditing purposes is the audio log showing the lack of synchronization, hence recording is mandatory.

When designing a system for broadcast monitoring, the above discussion justifies having an off-line design. Since recording is mandatory, the analysis of the audio can be done off-line, we can assume the stream is a collection of audio files. Even low tech companies with human listeners can analyze audio three times faster than real time, playing the recordings at a higher speed and skipping feature programming when tagging the audio logs. The human listener memorize a set of audio ads, afterwards, when playing the recording he/she identifies one of them and makes an annotation of the broadcast station log, writing the time of occurrence, and the ad ID. In this case accuracy of annotations lies within minutes. Human listeners can process 24 hours of audio in approximately 8 hours of work.

Our design replicates the above procedure in a digital way. We compare the audio-fingerprint of the stream with the corresponding audio-fingerprint of the audio ads being monitored. We then have annotations accuracy in the order of milliseconds, and 12 to 40 times faster than real time.

3.1 The Multi Band Spectral Entropy Signature

We describe in some detail the MBSES to put the reader in the appropriate context. The interested reader can obtain more information in references [9,10] and [14].

Obtaining the entropy of the signal directly in the time domain (more precisely the entropy of the energy of the signal) has proved to be very effective for audio-fingerprinting in [10]. With this approach, called *Time-domain Entropy Signature* (TES) the recall was high; but with some degradations, as equalization, it dropped quickly. To solve this problem in [9] the signal was divided in bands according to the Bark scale in the frequency domain, then entropy is determined for each band. The result was a very strong signature, with perfect recall even for strong degradations. Below we detail the extraction of the MBSES of an audio-signal.

1. The signal is processed in frames of 256 ms, this frame size ensures an adequate time support for entropy computation. The frames are overlapped by 7/8 (87.5%), therefore, a feature vector will be determined every 32 ms
2. To each frame the Hann window is applied and then its DFT is determined.
3. Shannon's entropy is computed for the first 21 critical bands according to the Bark scale (frequencies between 20 Hz and 7700 Hz). To compute Shannon's entropy, equation 1 is used. σ_{xx} and σ_{yy} also known as σ_x^2 and σ_y^2 are the variances of the real and the imaginary part respectively and $\sigma_{xy} = \sigma_{yx}$ is the covariance between the real and the imaginary part of the spectrum.

$$H = ln(2\pi e) + \frac{1}{2} ln(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2) \tag{1}$$

4. For each band obtain the sign of the derivative of the entropy as in equation (2). The bit corresponding to band b and frame n of the AFP is determined using the entropy values of frames n and $n-1$ for band b. Only 3 bytes for each 32 ms of audio are needed to store this signature.

$$F(n,b) = \begin{cases} 1\; if\;\; [h_b(n) - h_b(n-1)] > 0 \\ 0\; Otherwise \end{cases} \tag{2}$$

A diagram of the process of determining the MBSES of an audio-signal is depicted in Fig. 1.

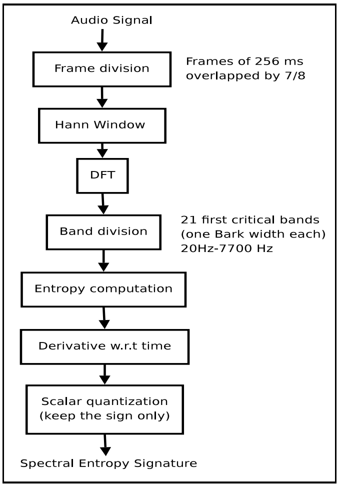

Audio Signal
Frame division
Frames of 256 ms overlapped by 7/8
Hann Window
DFT
Band division
21 first critical bands (one Bark width each) 20Hz-7700 Hz
Entropy computation
Derivative w.r.t time
Scalar quantization (keep the sign only)
Spectral Entropy Signature

Fig. 1. Computing the Spectral Entropy Signature

The fingerprint of the signal is now a binary matrix, with one column representing each frame in the signal. The most interesting part is that now the Hamming distance (the number of non matching bits compared element by element) is enough to measure similarity between signals.

3.2 The Monitoring Procedure

Monitoring is quite simple when we have a robust way to measure similarity between the audio stream and an audio segment (e.g once extracted the MBSES of both).

Figure 2 exemplifies the procedure for searching an occurrence of an ad in the stream. The smaller matrix (the audio ad) is slide one bit at a time to search for a match (a minimum in the distance).

We observed a peculiar phenomenon in searching for a minimum in the Hamming distance, there is a sudden increase just before there is a match, Figure 3 illustrated this, an ad was found in minutes 3 and 41. This is probably because the signature is not very repetitive, moreover, it is little compressible.

The Hamming distance can be efficiently computed with a lookup table counting the number of ones in a 21 bit string. This lookup table is addressed with the value of $x \oplus y$ with $\oplus$ the XOR operation between x and y the columns being compared.

4 Experiments

For our experiments we used all-day recordings from five different local AM (Amplitude Modulated) radio stations. This recordings were provided by *Contacto Media Research Mexico SA de CV* (CMR) in the lossy compression format

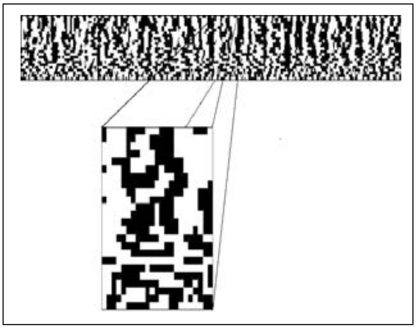


Fig. 2. The signature of the audio ad is the smaller matrix, the long grid is the signature of the monitored audio. When the Hamming distance falls below a threshold we count a match.

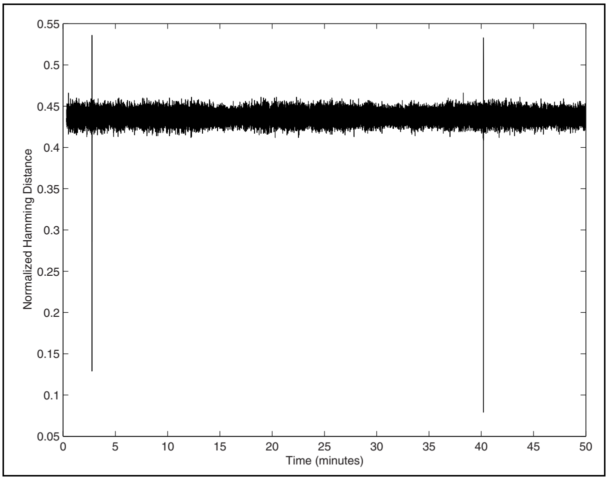


Fig. 3. This plot corresponds to the Hamming distance between the ad being searched and the corresponding segment in the audio stream. Notice a sudden increase followed by a decrease in the distance, both above and below a clear threshold.

mp3@64kbps spread in 95 files of approximately one hour each. Thirteen recordings of commercial spots were also provided to us as well as the results from manually monitoring these stations by their trained employees.

We determined the signatures of every one-hour mp3 file and stored them in separate binary files, generating 95 long signatures at this step. The process of checking all ad's occurrences in one-hour files lasted 40 seconds approximately. The whole process of checking 95 hours of audio generating the complete report took about an hour.

The report generated by our broadcast monitoring system was compared with the report provided by CMR. We found 272 occurrences while CMR reported only 231, the missed 41 ads were manually verified by us. It is noticeable that trained operators (human listeners) have failed to report those 41 spots, perhaps

Table 1. Comparison with the reported results on similar research

System	True positives rate (recognition rate)	False positives Rate (recognition mistakes rate)
Proposed System	100%	0%
Hellmuth et al [12]	99.8%	-
Oliveira et al [11]	95.4%	1%

due to fatigue or distraction. On the other hand all of the ad occurrences detected by operators were detected by our system.

The recognition rate reported by Hellmuth *et al* in [12] for similar experiments since they also use off-line monitoring, degrading by lossy compression precisely in the format mp3@64kbps and excerpts of 20 seconds (e.g the size of most commercial ads) was 99.8%. In contrast, our experiments report a precision of 100% since no commercial ad occurrence was missed with our system. Table 1 compares this results including the results reported by Oliveira *et al* in, [11].

5 Conclusions and Future Work

We found our Multi-band spectral entropy signature (MBSES) to be adequate for robust automatic radio broadcast monitoring. The time resolution of the signature was adjusted to work with commercial spots with high speech content.

Instead of searching sequentially among the collection of spots for an occurrence of any of them, we will design a proximity index that would allow working with thousands of spots without affecting the speed of the monitoring process. On the other hand, preliminary results about using *graphic processing units* (GPU) for computing the fingerprint shows an important speedup with respect to single core computing. This also pose very interesting audio mining challenges in archived audio logs of several-year long recordings.

Acknowledgements

We want to thank the firm *Contacto Media Research Services SA de CV* in Guadalajara, México for providing us with the manually tagged recordings used in this paper. We wish to thank the comments and suggestions of the anonymous referees helping to improve the presentation.

References

1. Haitsma, J., van der Veen, M., Kalker, T., Bruekers, F.: Audio watermarking for monitoring and copy protection. In: MULTIMEDIA 2000: Proceedings of the 2000 ACM workshops on Multimedia, pp. 119–122. ACM, New York (2000)
2. Nakamura, T., Tachibana, R., Kobayashi, S.: Automatic music monitoring and boundary detection for broadcast using audio watermarking. In: SPIE, pp. 170–180 (2002)

3. Sigurdsson, S., Petersen, K.B., Lehn-Schioler, T.: Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. In: International Symposium on Music Information Retrieval, ISMIR (2006)
4. Logan, B.: Mel frequency cepstral coefficients for music modeling. In: International Symposium on Music Information Retrieval, ISMIR (October 2000)
5. Herre, J., Allamanche, E., Hellmuth, O.: Robust matching of audio signals using spectral flatness features. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 127–130 (2001)
6. Hellman, R.P.: Asymmetry of masking between noise and tone. Perception and Psychophysics 11, 241–246 (1972)
7. Pauws, S.: Musical key extraction from audio. In: International Symposium on Music Information Retrieval ISMIR, October 2004, pp. 96–99 (2004)
8. Cano, P., Battle, E., Kalker, T., Haitsma, J.: A review of algorithms for audio fingerprinting. In: IEEE Workshop on Multimedia Signal Processing, pp. 169–167 (2002)
9. Camarena-Ibarrola, A., Chávez, E.: On musical performances identification, entropy and string matching. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 952–962. Springer, Heidelberg (2006)
10. Camarena-Ibarrola, A., Chávez, E.: A robust entropy-based audio-fingerprint. In: Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, ICME, pp. 1729–1732. IEEE CS Press, Los Alamitos (2006)
11. Oliveira, B., Crivellaro, A., César Jr., R.M.: Audio-based radio and tv broadcast monitoring. In: WebMedia 2005: Proceedings of the 11th Brazilian Symposium on Multimedia and the web, pp. 1–3. ACM Press, New York (2005)
12. Hellmuth, O., Allamanche, E., Cremer, M., Kastner, T., Neubauer, C., Schmidt, S., Siebenhaar, F.: Content-based broadcast monitoring using mpeg-7 audio fingerprints. In: International Symposium on Music Information Retrieval ISMIR (2001)
13. Group, M.A.: Text of ISO/IEC Final Draft International Standard 15938-4 Information Technology - Multimedia Content Description Interface - Part 4: Audio (July 2001)
14. Camarena-Ibarrola, J.A.: Identificación Automática de Señales de Audio. PhD thesis, Universidad Michoacana de San Nicolás de Hidalgo (January 2008)

Real Time Hot Spot Detection Using FPGA

Sol Pedre, Andres Stoliar, and Patricia Borensztejn

Departamento de Computación, Facultad de Ciencias
Exactas y Naturales, Universidad de Buenos Aires
{spedre,astoliar,patricia}@dc.uba.ar

Abstract. Many remote sensing applications require on-board, real time processing with low power consumption. Solutions based in FPGA implementations are common in these cases to optimize the processing resources needed. In this paper we describe an FPGA based solution for a remote sensing application that processes real time video from an infrared camera in order to identify hot spots. The solution reduces the information in each frame to the location and spatial configuration of each hot spot present in the frame. The proposed method successfully segments the image with a total processing delay equal to the acquisition time of one pixel (that is, at the video rate). This processing delay is independent of the image size. The solution is not tied up to one specific camera, and may be used with several infrared cameras with minor adjustments. FPGA area equations are also presented in order to calculate the needed FPGA size for a particular application.

Keywords: real time image processing, FPGA, remote sensing, hot spot detection, embedded computing.

1 Introduction

Many remote sensing applications require on board, real time processing with low power consumption. For many of these embedded digital signal processing applications, today´s general purpose microprocessors are not longer able to handle them [1]. Functional Programming Gate Array (FPGA) offer a highly parallel hardware architecture with low power consumption that is an alternative for such digital signal processing implementations.

Field Programmable Gate Arrays, or FPGA, are devices made up of thousands of logic cells and memory. Logic cells, memory and interconnections between them are software programmable using a standard computer. Therefore, these devices offer a fast and cheap prototype solution for an embedded product, and when the production scale is small, they also offer a fine final solution.

In this paper we present and describe a real time image processing algorithm, hot spot detection, implemented on an FPGA device. This solution was developed for an Unmanned Aerial Vehicle (UAV) System of the Department of Computer Architecture, Escola Politècnica Superior de Castelldefels, Universitat Politècnica de Catalunya. The aim of the algorithm is to identify fire embers (that is, hot spots) in the images captured by an infrared video camera on the UAV. The location and characteristics of

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 595–602, 2009.

the detected hot spots are then transmitted by the UAV to a firemen team fighting a forest fire [2]. Our solution is general enough to be integrated in other systems, and with different cameras. The rest of this paper is organized as follows: section 2 explains the hot spot detection algorithm, section 3 explains the FPGA implementation, section 4 shows the experiments and results and section 5 explains some conclusions.

2 Hot Spot Detection

The problem consists on processing video as it is captured by an IR camera on an Unmanned Aerial Vehicle. The UAV has a network centric architecture, in which all sensors and different processing units are connected to an Ethernet network [2]. Our proposed FPGA solution is inserted between the IR camera and the network. It takes the analogical output of the IR camera, processes the frames in real time and returns the location and spatial configuration of the found hot spots (if any) in UDP packets. As many infrared cameras have analogical outputs, as composed video, the proposed solution can be used for different IR cameras and is not tied up to one specific camera.

The most important constraint for the solution is that the image needs to be processed in real time, with the minimum possible delay between the acquisition of the last pixel and the transmission of the results. It is also desirable that the whole application works at the slowest possible clock frequency, to minimize the FPGA's power consumption.

In order to fulfill these requirements, the proposed algorithm and hardware implementation exploit the intrinsic parallelism of the process, obtaining the results of a complete frame at the moment that its acquisition is finished, with a total processing delay equal to the acquisition time of one pixel (i.e, the camera´s pixel video frequency). Moreover, as the camera delivers continuous images, the results of the previous frame are transmitted in parallel with the processing of the current frame. Finally, the application runs with the smallest possible clock frequency that allows to fulfill the previous requirements: the IR camera´s pixel clock frequency. This also simplifies the integration between the camera and the proposed solution.

2.1 Segmentation Algorithm

The proposed algorithm segments the image in hot and cold regions, storing the location and spatial configuration of the found hot regions (i.e, hot spots). Its complexity lies in grouping the pixels in hot spots and updating the stored hot spot's data as the image is being captured. The IR camera's output video is first digitalized, the temperature pixel is extracted and then classified as a hot o cold pixel (i.e, if the pixel belongs to a hot spot or not). The segmentation algorithm then checks if the adjacent pixels belong to hot spots and decide if the current pixel is the beginning of a new hot spot, if it belongs to an already existing hot spot, and whether this pixel unifies two previously discovered hot spots. It also updates the stored hot spot´s data accordingly.

In this manner, the algorithm performs the segmentation of the image using only the current pixel and a list *L* that stores to which hot spot (if any) the previous line of pixels belong to. Therefore, there is no need for extra memory to store parts or the complete image, and the total processing delay is independent on the image size. The algorithm´s pseudo code is shown in *Listing1.*

Listing 1. Segmentation Algorithm

```
inputs:
- pixel(m,n)
- line L of previous pixels indicating which hot spot the belong to, if any.

receive pixel(m,n)
    if pixel(m,n) does not belong to a hot spot
            mark in the line L that pixel(m,n) does not belong to a hot spot.
    if pixel(m,n) belongs to a hot spot
        if pixel(m-1,n) and pixel(m,n-1) do not belong to hot spots
                create a new hot spot for pixel(m,n)
                mark pixel(i,j) in the line L as belonging to the new hot spot
        if (pixel(m-1,n) belongs to hot_spot_x and pixel(m,n-1) does not belong to any hot spot)
        or (pixel(m-1,n) does not belong to any hot spot and pixel(m,n-1) belongs to hot_spot_x)
        or (pixel(m-1,n) and pixel(m,n-1) belong to hot_spot_x))
                add pixel(m,n) to hot_spot_x in the memory
                mark pixel(m,n) in line L as belonging to hot_spot_x
        if (pixel(m-1,n) belongs to hot_spot_x and pixel(m, n-1) belongs to hot_spot_y
        and id(hot_spot_x) < id(hot_spot_y))
                add hot_spot_y data to hot_spot_x  in the memory
                add pixel(m,n) to hot_spot_x in the memory
                mark  hot_spot_y  as invalid in the memory
                mark pixel(m,n) in line L as belonging to hot_spot_x
                for each pixel in line L
                        if (pixel belongs to hot_spot_y)
                                mark pixel as belonging to hot_spot_x
```

3 FPGA Implementation

We propose the use of FPGA technology to achieve the real time processing of the image, with a total processing delay equal to the acquisition time of one pixel (i.e video frequency). Some hardware architectures have been presented in literature aimed at accelerating image processing methods [3][4][5][6] but none intended for the segmentation of an IR image for hot spot detection.

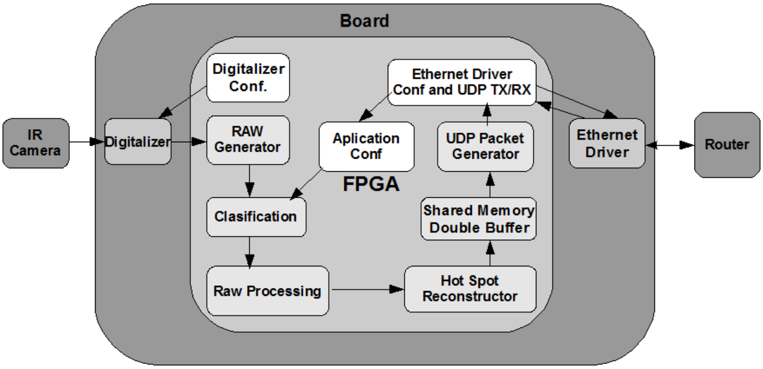


Fig. 1. Complete board including the video digitalizer, the FPGA and Ethernet physical driver

In this section we describe the FPGA implementation, focusing in the Raw Processing, Hot Spot Reconstructor and Double Buffer Shared Memory modules that are the core of the solution. These modules implement the proposed algorithm (*Listing 1*) in a way that the real time constraint is achieved. Fig. 1 shows the complete hardware solution.

3.1 Raw Processing, Hot Spot Reconstructor and Memory Modules

The Raw Processing module determines if the current pixel is the beginning of a new hot spot, if it belongs to an already existing hot spot, and whether this pixel unifies two previously discovered hot spots. To do this. it keeps the line *L* of previous pixels showing to which hot spot they belong to, and update this line as shown in *Listing1*.

The core of the Raw Processing module is the implementation of *L* such as to calculate which hot spot the current pixel belongs to and update all the list in only one pixel clock. For this purpose, *L* is implemented as a stack: the top of the stack stores the *id* of the hot spot that *pixel(m,n-1)* belongs to, and the bottom of the stack stores the *id* of the hot spot *pixel(m-1,n)* belongs to. This two special records can be accessed to obtain the hot spot *ids* needed in the algorithm. The remaining records in *L* hold the information of the line of pixels between *pixel(m-1,n)* and *pixel(m,n-1)*, i.e, the previous line of pixels. In each clock, the hot spot *id* corresponding to the new pixel is pushed onto the stack, all the middle records are updated if necessary and moved to the next stack position, and the bottom record (i.e, the hot spot *id* of *pixel(m-1,n)*) is discarded.

When a pixel unifies two previously discovered hot spots, say *hot_spot_y* and *hot_spot_x*, *hot_spot_y* is marked as invalid and all the pixels belonging to that hot spot are added to *hot_spot_x*. In that case, all records in the stack have to be accessed, compared with the *id* of *hot_spot_y* and changed to the *id* of *hot_spot_x* (if needed) in one clock cycle. To accomplish this, each record of the stack has a comparator and a multiplexer. The result of comparing the record's *id*, say *idA*, with the *id* of *hot_spot_y* enables the multiplexer that either propagates *idA* or the *id* of *hot_spot_x* to the next record as needed. The implementation of one record of the list *L* in this module is shown in Fig. 2.

The Hot Spot Reconstructor module is in charge of updating the information of hot spots found to the moment with the information from the current pixel, as shown in the algorithm in *Listing 1*. The Raw Processing module tells the Hot Spot Reconstructor module whether it has to create a new hot spot with the current pixel, unify two hot spots or simply add the pixel to an existing hot spot. This module has to access the hot spot Memory, retrieve the corresponding information, recalculate the data and write the results back, all in one pixel clock. The implementation of this module is shown in Fig. 3

The Memory module is designed as a shared double buffer. Each buffer is organized as a vector of records, with one record for each hot spot. Each record stores the location and spatial configuration of the hot spot. The partial results of the current frame are stored in one buffer, while the final results of the previous frame are stored in the other one. In this manner, the results of the previous frame can be transmitted in parallel with the segmentation of the current frame.

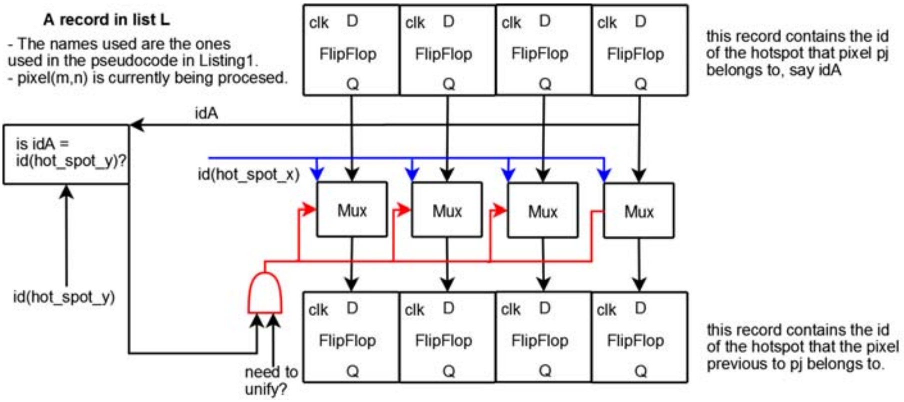

A record in list L
- The names used are the ones used in the pseudocode in Listing1.
- pixel(m,n) is currently being procesed.
clk D FlipFlop Q
clk D FlipFlop Q
clk D FlipFlop Q
clk D FlipFlop Q
this record contains the id of the hotspot that pixel pj belongs to, say idA
idA
is idA = id(hot_spot_y)?
id(hot_spot_x)
Mux
Mux
Mux
Mux
id(hot_spot_y)
need to unify?
clk D FlipFlop Q
clk D FlipFlop Q
clk D FlipFlop Q
clk D FlipFlop Q
this record contains the id of the hotspot that the pixel previous to pj belongs to.

Fig. 2. Implementation of a record of list *L* in the Raw Processing module

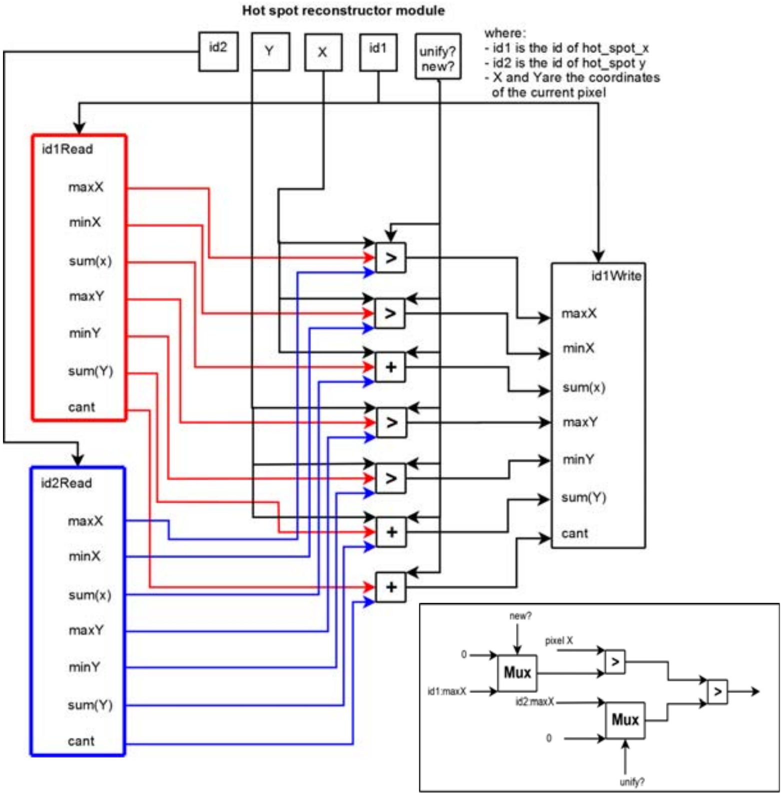


Fig. 3. Hot Spot Reconstructor module and detail of the logic for the maximum calculation

3.2 Auxiliary Modules

The Raw Generator module generates the RAW stream by extracting the temperature pixels from the digitalized video. The Classification module classifies each pixel as belonging to a hot spot or not, depending on its IR radiation determined by the pixel´s value.

The UDP Packet Generator is the module in charge of creating the correct UDP packets with the hot spot data of the previous frame. As this module works with the clock frequency of the MAC Ethernet Module and the rest of the application works with a different clock frequency (i.e, the IR camera´s pixel clock frequency), we use a FIFO to solve the associated problems with the exchange of information between two different clock domains.

Finally, the configuration modules allows to configure the integrated circuits on the hardware board outside the FPGA: the SAA7113 digitalizer and the Ethernet physical driver. There is also an Application Configuration module that allows to configure through the Ethernet connection variables such as the threshold for the classification module or the ip address and port of the UAV's CPU.

3.3 Solution Sizing

In order to implement the high parallelism needed to achieve the proposed real time processing of the image, much space and hardware resources of the FPGA are used. The area needed for this implementation depends only on the size of the image and the maximum amount of hot spots that can be found in each image. In order to make the application suitable for different IR cameras, those parameters can be easily configured.

There are two modules in the implementation that are resource consuming: the Raw Processing module and the Memory module. The Raw Processing module implements the list *L* that stores the hot spot *id* for each pixel in the previous line, as explained in section 3.1. In terms of FPGA area, the list has *im_width* records. Each record is wide enough to store a hot spot *id*, that is *log(max_hotspot_amount)* bits, and has extra logic needed for the hot spot unifying process. The area equations is as follows:

$$\mathit{im\_width}*\ [\log(\mathit{max\_hotspot\_amount})*(1\ \text{flip flop} + 1\ \text{mux}) + 1\ \text{comparator of}\ \log(\mathit{max\_hotspot\_amount})\ \text{bits}]$$

The Memory module stores the information of each hot spot found in the image, that is, the memory has *max_hotspot_amount* records. Each record stores the information needed to calculate the location and spatial configuration of one hot spot, that is: {*maxX, minX, sumX, maxY, minY, sumY, count of pixels}*. Finally, there are two memory buffers, one with the final results of the previous frame and one with the partial results of the current frame. The hot spot memory is implemented using the FPGA block rams. The needed block ram equation is as follows:

$$2*\mathit{max\_hotspot\_amount}*\ [\log(\mathit{im\_width})+\log\ (\mathit{im\_width}) + (2*\log(\mathit{im\_width})-1) + \log(\mathit{im\_height})+\log\ (\mathit{im\_height})+(2*\log(\mathit{im\_height})-1)+\log(\mathit{im\_width}*\mathit{im\_height})-1]$$

From these equations, we can see that the amount of logic cells needed depend linearly on the image width, while the amount or memory (block rams) depends linearly on the maximum amount of hot spots to be detected per frame. The image height is of little importance for area calculations. With these equations, it is straightforward to calculate the size of the needed FPGA given the size of the image and a maximum for the amount of hot spots expected in each frame.

4 Experiments and Results

Our testing environment consists of a PAL-N composed video camera, a development kit with a Xilinx Virtex 4 FX12-10C-ES FPGA, a SAA7113 video digitalizer and an Ethernet physical driver. The development environment used is the Xilinx ISE Webpack 10.1, with default settings for all the involved processes. The UDP packets with the resulting hot spots are routed to a PC in order to check the processing results. From the video camera we process and analyze 50 frames per second, corresponding to half video images even and odd, with 512 pixels by 256 lines per frame. The pixel clock generated by the SAA7113 digitalizer is the standard 27 Mhz clock for video coding ITU-R BT 656 YUV 4:2:2. The solution was configured for a 256 maximum amount of hot spots per frame. In Fig. 4 some results are shown.

As all the memories were mapped into block rams, including the UDP FIFO and the configuration memory for the SAA7113, and the hardware MAC Ethernet included in the Virtex 4 was used, the total area of the application was 84% of the FPGA slices (1 LUT + 1 flip flop) and 32% of the block rams.

In particular, the occupied area of the main modules (Raw Processing, Hot Spot Reconstructor and Memory) was 79% of the FPGA slices and 25% of the block-rams. This area corresponds to the size equations presented in section 4, and it means that the entire application fits in the smallest Virtex 4 FPGA available in the market. The maximum operation clock obtained was of little over 100 Mhz, which is enough to work with the 27 Mhz clock output of the SAA7113. In the UAV application, the mounted IR camera is the FLIR A320, that delivers 320 pixels by 240 lines images at a rate of 9 fps. Therefore, the results of the tests in the laboratory experiments indicate that the solution is well suited for the UAV application.

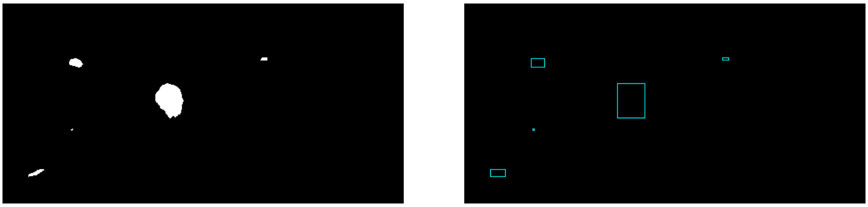


Fig. 4. *left:* hot spot image after classification in hot or cold pixels. *right:* visual representation of the results showing the location of the detected hot spot (the center of mass is not shown).

5 Conclusions

In this paper we proposed the use of FPGA technology to achieve real time processing of an IR image for hot spot detection. The proposed method successfully segments the image with a total processing delay equal to the acquisition time of one pixel (that is, at the video rate). This processing delay time is independent of the image size. There is also no need for extra memory to store parts or the complete image. The proposed solution is not tied to one specific IR camera, and may be used with several IR camera with minor adjustments. FPGA area equations were presented in order to calculate the needed FPGA size for a particular application.

The experiments show that the maximum operation clock is of little over 100 Mhz, which is enough to work with the 27 Mhz clock output of the SAA7113. Moreover, the entire application fits into the smallest Virtex 4 FPGA available in the market. The results also show that the proposed method is well suited to work with the FLIR A320 camera on the UAV application.

References

1. Dally, W., Balfour, J., Black-Shaffer, D., Chen, J., Chen, H.C., Parikh, V., Park, J., Sheffield, D.: Efficient Embedded Computing. Computer, 27–32 (2008), IEEE 0018-9162
2. Salami, E., Pedre, S., Borenzstejn, P., Barrado, C., Stoliar, A., Pastor, E.: Decision Support System for Hot Spot Detection. In: 5th International Conference on Intelligent Environments (IE 2009), Universitat Politecnica de Catalunya, Barcelona, Spain (2009)
3. Draper, B.A., Beveridge, R., Willem, B.A.P., Ross, C., Chawathe, M.: Accelerated image processing on FPGAs. IEEE Transactions on Image Processing 12(12), 1543–1551 (2003)
4. Torres, C., Arias, M.: FPGA-based Configurable Systolic Architecture for Windows-based Image Processing. EURASIP Journal on Applied Signal Processing, Special Issue on Machine Perception in a Chip 2005(7), 1024–1034 (2005)
5. Chang, C., Hsiao, P., Huang, Z.: Integrated Operation of Image Capturing and Processing in FPGA. IJCSNS International Journal of Computer Science and Network Security 6(1), 173–180 (2006)
6. Chang, L., Rodés, I., Méndez, H., del Toro, E.: Best-Shot Selection for Video Face Recognition Using FPGA. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 543–550. Springer, Heidelberg (2008)

Fast Pattern Classification of Ventricular Arrhythmias Using Graphics Processing Units

Noel Lopes[1,2] and Bernardete Ribeiro[1]

[1] CISUC - Center for Informatics and Systems of University of Coimbra, Portugal
[2] UDI/IPG - Research Unit, Polytechnic Institute of Guarda, Portugal
noel@ipg.pt, bribeiro@dei.uc.pt

Abstract. Graphics Processing Units (GPUs) can provide remarkable performance gains when compared to CPUs for computationally-intensive applications. In the biomedical area, most of the previous studies are focused on using Neural Networks (NNs) for pattern recognition of biomedical signals. However, the long training times prevent them to be used in real-time. This is critical for the fast detection of Ventricular Arrhythmias (VAs) which may cause cardiac arrest and sudden death. In this paper, we present a parallel implementation of the Back-Propagation (BP) and the Multiple Back-Propagation (MBP) algorithm which allowed significant training speedups. In our proposal, we explicitly specify data parallel computations by defining special functions (*kernels*); therefore, we can use a fast evaluation strategy for reducing the computational cost without wasting memory resources. The performance of the pattern classification implementation is compared against other reported algorithms.

Keywords: GPU Computing, Parallel Programming, Neural Networks.

1 Introduction

Neural networks (NNs) have been successfully applied as pattern recognition systems in many areas [1,2]. However building a NN solution, is usually a computationally expensive task, demanding a considerable amount of time. Depending on the complexity of the problem, in most cases several NNs, with different configurations, must be trained before achieving a good solution. Thus the time required to train the NNs alone may prevent high quality solutions from being found. Dedicated hardware can be used to overcome this problem. Nevertheless this solution is often not chosen due to its high cost and reduced flexibility [1].

Graphics Processing Units (GPUs) can offer a more flexible and economical alternative to the use of dedicated hardware, yet a powerful one. Originally GPUs were developed as specialized accelerators for triangle rasterization. The transition to general purpose engines, aiming at high throughput floating-point computation, is witnessed by GPU implementations of machine learning algorithms [3,4]. At present computers possess a GPU that offers increasing degrees of programmability allowing enough flexibility to be used to accelerate non-graphics applications [5]. GPUs are much more effective in utilizing parallelism

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 603–610, 2009.

(and pipelining) than general purpose CPUs [6]. Due to its inherent parallel architecture, GPUs can provide remarkable performance gains when compared to CPUs for computationally-intensive applications. Thus they provide an attractive alternative to use dedicated hardware in machine learning, namely in the NN field [5]. Moreover, GPUs still service the large gaming industry and so they are relatively inexpensive [3]. However, until recently, General-Purpose computation on the GPU (GPGPU), required the programmer to master the fundamentals of graphics shading languages that require prior knowledge on computer graphics [6]. This changed when NVIDIA introduced a new data-parallel, C-language programming API called CUDA (Compute Unified Device Architecture) that bypasses the rendering interface and avoids the difficulties of classic GPGPU [4,6].

The rest of this paper is organized as follows. Section 2 introduces the CUDA programming model and its architecture. Section 3 summarizes the Multiple Back-Propagation (MBP) algorithm, whose GPU parallel implementation is discussed latter on section 4. Section 5 details the steps taken to create NNs capable of detecting VAs based on time and frequency domain features obtained from electrocardiography's (ECGs). Section 6 compares the GPU and CPU implementations of the algorithms. Finally, section 7 summarizes contributions and addresses directions for future work.

2 CUDA Programming Model and Architecture

The CUDA programming model extends the C language, allowing the programmer to explicitly specify data parallel computations by defining special functions, named *kernels*. *Kernels* are executed in parallel by different CUDA threads, on a physically separate device (GPU) that operates as a co-processor to the host (CPU) running the program. Figure 1 shows an example of a simple *Kernel*. Threads are organized into blocks that are required to execute independently. To invoke a *kernel*, programmers use language extensions in order to specify the runtime values for the number of blocks (organized into a two dimensional grid) and the number of threads per block [7]. The CUDA programming model is supported by an architecture built around a scalable array of multi-threaded Streaming Multiprocessors (SMs). Each SM has eight Scalar Processor (SP) cores. When a program on the host invokes a *kernel* grid, its blocks are enumerated and distributed to SM with available execution capacity. As thread blocks finish their execution, new blocks are launched on the vacated SMs. Each

```
__global__ void WI(float * w, float * i, float * o, int size) {
       int idx = blockIdx.x * blockDim.x + threadIdx.x;
       if (idx < size) o[idx] = w[idx] * i[idx];
}
```

Fig. 1. *Kernel* that multiplies each element of the vector **w** by the corresponding element of the vector **i** placing the result on the vector **o**

SM creates, manages, and executes concurrent threads in hardware with zero scheduling overhead and can implement fast barrier synchronization. These are the keys to efficiently support fine-grained parallelism [7].

3 Multiple Back-Propagation

Multiple Back-Propagation (MBP) is a generalization of the Back-Propagation (BP) algorithm that can be used to train Multiple Feed-Forward (MFF) networks [8]. Jointly MFF networks and the MBP algorithm shape an architecture that is (in most situations) preferable to the use of feed-forward (FF) networks trained with the BP algorithm. MFF networks are obtained by integrating two FF networks (a main network and a space network) as shown in Figure 2 [9]. The main network contains at least one selective activation neuron. Selective activation neurons differentiate between *stimulus* (patterns). Their response depends on the space localization of a pattern p presented to the network and might be amplified or reduced accordingly. Its output is given by (1):

$$y_k^p = m_k^p \mathcal{F}_k(a_k^p) = m_k^p \mathcal{F}_k(\sum_{j=1}^{N} w_{jk} y_j^p + \theta_k) \ , \tag{1}$$

where y_k^p is the output of neuron k, m_k^p the importance of the neuron, that varies accordingly to the pattern (*stimulus*) presented to the network, $\mathcal{F}_k$ the neuron activation function, a_k^p its activation, θ_k the bias and w_{jk} the weight of the connection between neuron j and neuron k. The importance (m_k^p) of each neuron k for the current pattern p is determined by a standard FF network, that receives the same inputs as the main network, named space network because it is implicitly dividing the input space. The main network can only calculate its outputs after knowing the outputs (m_k^p) of the space network. Thus the two networks will function in a collaborative manner and must be trained together.

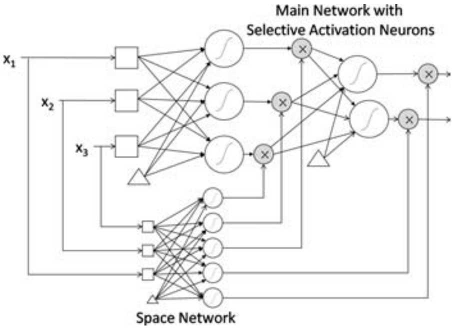


Fig. 2. MFF Network. Squares represent input neurons, white circles hidden and output neurons, gray circles multipliers and triangles the bias.

4 BP and MBP Parallel GPU Implementation

In this phase we choose to implement exclusively the batch training mode for both the BP and MBP algorithms, since this is the mode that can benefit the most from a parallel implementation. The resulting implementation exploits the widely used adaptive step size technique due to stability of the algorithms and improved training speeds. In order to simultaneously implement both algorithms, five *kernels* listed on Table 1 were created. Figure 3 illustrates the steps necessary to train, during one epoch, a MFF network comprising a main network containing an input layer with N_i neurons, a hidden layer with N_h neurons with selective activation and an output layer with N_o neurons (without selective activation) and a space network containing an input layer with N_i neurons and an output layer with N_h neurons. This is the configuration we will use later on section 6 to train MFF networks for the VA problem. It is assumed the training set contains N_p patterns. First FireLayer is called with the network inputs vector $\mathbf{x}$ and the weights vector of the space network $\mathbf{w_s}$. As a result a vector $\mathbf{m}$ containing the importance of each neuron with selective activation, for each pattern, is calculated. This vector is then used together with $\mathbf{x}$ and the vector containing the input weights of the main network hidden layer $\mathbf{w_h}$ to call FireLayer in order to calculate the hidden layer outputs $\mathbf{y_h}$. To complete the calculation of the network outputs $\mathbf{y}$, FireOutputLayer is then called using $\mathbf{y_h}$, the input weights of the output layer $\mathbf{w_o}$ and the desired outputs vector $\mathbf{d}$. This will also calculate the local gradients of the output layer $\delta_{\mathbf{o}}$. At this time the Root Mean Square (RMS) error of the network can be calculated by calling the CalculateRMS *kernel*. Then CalcLocalGradients is used to determine the local gradients of the hidden layer $\delta_{\mathbf{h}}$ and the local gradients of the space layer $\delta_{\mathbf{s}}$. Finally CorrectWeights is called several times to adjust the weights $\mathbf{w_s}$, $\mathbf{w_h}$ and $\mathbf{w_o}$ of the MFF network. Figure 4 shows the percentage of time spent by the GPU in each *Kernel* for the VA problem.

Table 1. *Kernels* used to implement both the BP and the MBP algorithms. The *Kernels* shown here process all the training patterns simultaneously.

Kernel	**Purpose**
FireLayer	Calculates the outputs of all neurons in a given layer.
FireOutputLayer	Calculates the outputs of the NN output layer. If the layer contains selective activation neurons, the local gradients of the corresponding space network neurons are also calculated.
CalcLocalGradients	Calculates the local gradient of all neurons in a hidden layer. If there are selective activation neurons, the local gradients of the corresponding space network neurons are also calculated.
CorrectWeights	Adjust the weights of a given layer.
CalculateRMS	Calculate the Root Mean Square (RMS) error of the network.

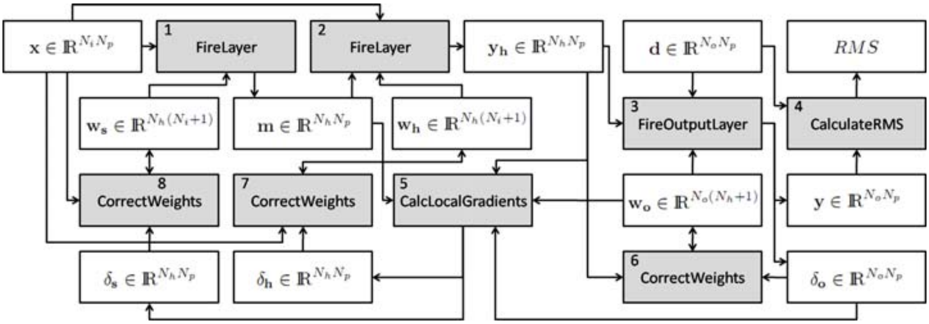

$\mathbf{x} \in \mathbb{R}^{N_i N_p}$
1 FireLayer
2 FireLayer
$\mathbf{y_h} \in \mathbb{R}^{N_h N_p}$
$\mathbf{d} \in \mathbb{R}^{N_o N_p}$
RMS
$\mathbf{w_s} \in \mathbb{R}^{N_h(N_i+1)}$
$\mathbf{m} \in \mathbb{R}^{N_h N_p}$
$\mathbf{w_h} \in \mathbb{R}^{N_h(N_i+1)}$
3 FireOutputLayer
4 CalculateRMS
8 CorrectWeights
7 CorrectWeights
5 CalcLocalGradients
$\mathbf{w_o} \in \mathbb{R}^{N_o(N_h+1)}$
$\mathbf{y} \in \mathbb{R}^{N_o N_p}$
$\delta_\mathbf{s} \in \mathbb{R}^{N_h N_p}$
$\delta_\mathbf{h} \in \mathbb{R}^{N_h N_p}$
6 CorrectWeights
$\delta_\mathbf{o} \in \mathbb{R}^{N_o N_p}$

Fig. 3. Model of the sequence of *kernels* launched by the host (in each epoch) to train a MFF network comprising a main network with 3 layers and a space network with 2 layers (that calculates the importance of the hidden neurons of the main network). White rectangles represent input and output vectors, whilst gray rectangles represent *kernels*.

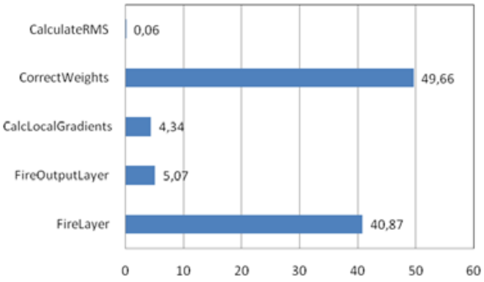
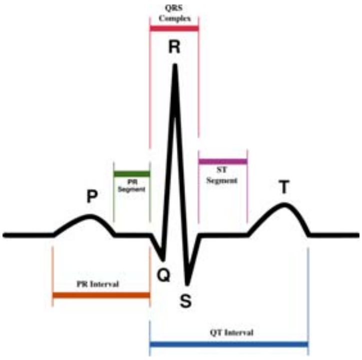


Fig. 4. Percentage of time spent by the GPU in each *Kernel*

5 Ventricular Arrhythmias Assessment

Nowadays most countries face high and increasing rates of cardiovascular diseases. In Portugal there is a 42% probability of dying of these diseases and worldwide they are accountable by 16.7 million deaths per year [10]. In this context VAs assume a significant role given that their prevalence can lead to life threatening conditions that may result in cardiac arrest and sudden death. VAs evolve from simple Premature Ventricular Contractions (PVCs) which are usually benign, to ventricular tachycardia and finally to critical ventricular fibrillation episodes which are potentially fatal and the main cause of sudden cardiac death. The detection of PVCs from an ECG is thus of major importance, since they are associated with an increased risk of adverse cardiac events. A typical ECG tracing of a ordinary heartbeat consists of a P wave, a QRS complex and a T wave (see figure 5a). PVCs result from an ectopic depolarization on the ventricles, which results on a wider and abnormally shaped QRS complex. Moreover, typically QRS complexes are not preceded by P waves, and T waves are usually larger and with opposite deflection to the QRS complex. For high-performance detection of VAs we take advantage of the power of GPUs to significantly



Feature	Description
RRav	RR mean interval
RR0	Last RR interval
SN	Signal/Noise estimation
Ql	Q-wave length
(Qcx, Qcy)	Q-wave mass center (x,y) coordinates
(Qpx, Qpy)	Q-wave peak (x,y) coordinates
Rl	R-wave length
(Rcx, Rcy)	R-wave mass center (x,y) coordinates
(Rpx, Rpy)	R-wave peak (x,y) coordinates
Sl	S-wave length
(Scx, Scy)	S-wave mass center (x,y) coordinates
(Spx, Spy)	S-wave peak (x,y) coordinates

Fig. 5. (a) Schematic diagram of normal sinus rhythm for a human heart as seen on ECGs. (b) Selected features from the ECG signal.

accelerate the training of a NN based approach. The NNs take as inputs 18 features (see figure 5b) that were chosen in [11,12]. For comparison, we also use the same training, test and validation datasets, each one containing 19391 samples.[1]

6 Results for the CPU and GPU Implementations

In order to compare the performance of the GPU and the CPU versions we respectively used (*i*) the proposed CUDA implementation (see section 4) and (*ii*) the Multiple Back-Propagation software. Multiple Back-Propagation is a highly optimized software, developed in C++, for training NNs with the BP and MBP algorithms.[2] The GPU version was benchmarked on two different NVIDIA devices: a GeForce 8600 GT with 4 SM (32 cores) and a GTX 280 with 30 SM (240 cores). The CPU version was benchmarked on a Intel Core 2 6600 CPU (2.4 GHz). Results were obtained, using the VA datasets, both for the BP and the MBP algorithms. The FF networks trained consisted of 3 layers. As for MFF networks the topology was described in section 4 (see Figure 3 description). Experiments demonstrate that the GPU implementation delivers considerable speedups comparatively to the CPU. Figure 6 shows the number of epochs trained per minute accordingly to the hardware. Using the GTX 280 GPU it is possible to reduce the training time more than 50 times relatively to the CPU, as shown in Figure 7. It is interesting to note that as the number of hidden neurons increases so does the gain of speed provided by the GPU, because more processing that can be parallelized is required. Since currently our implementation does not support cross validation, preliminary tests were conducted in order to determine when to stop training. Based on the information collected we decided to train both FF and MFF networks during 1 million

[1] MIT-BIH Arrhythmia Database (http://www.physionet.org/physiobank/)

[2] Multiple Back-Propagation software is freely available at http://dit.ipg.pt/MBP

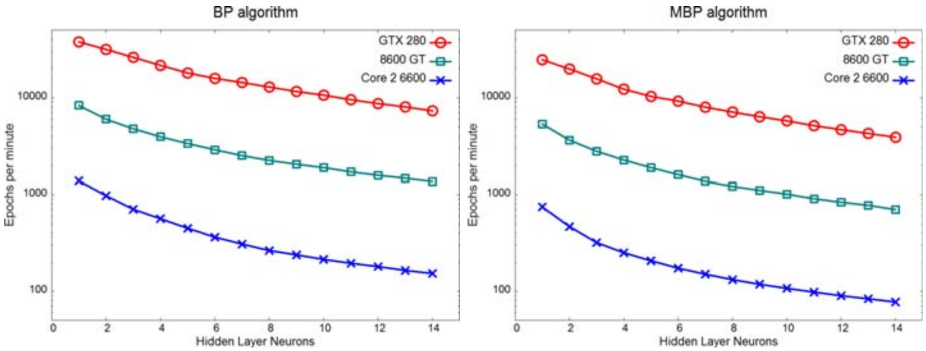


Fig. 6. Number of epochs trained per minute, accordingly to the hardware

Table 2. PVC detection: performance results of the NNs

Metrics	BP (FF)			MBP (MFF)		
	Train	Test	Val	Train	Test	Val
Sensitivity	98.07	95.94	94.67	97.42	95.54	94.47
Specificity	99.84	99.62	99.61	99.87	99.68	99.70
Accuracy	99.70	99.33	99.23	99.68	99.36	99.30

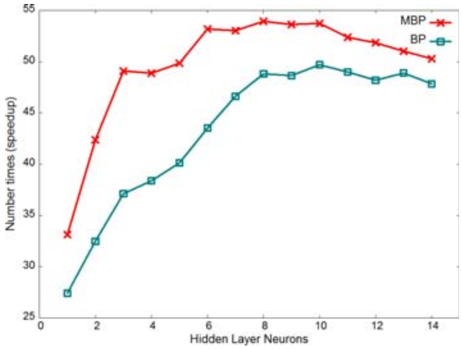


Fig. 7. Increase in speed provided by the GTX 280 relatively to the CPU

epochs, varying the number of hidden neurons. It is worth to mention that during the preliminary tests some NN were trained up to 3 million epochs, requiring almost 9 hours of train on a GTX 280 GPU. We estimate that if such NNs were trained on the CPU we would need almost 3 weeks to train each one. Table 2 shows the performance results of the best networks found, trained with the BP and the MBP. The best network trained with the BP algorithm has 14 hidden neurons and the best trained with the MBP has 13 hidden neurons with selective activation. The results, which improve over those previously obtained in [11,12], could not be obtained without the gain of speed provided by the GPU.

7 Conclusion

In this paper, the parallel implementation of MBP (and BP) algorithms to train MFF neural networks has proved highly efficient for pattern classification. Results confirm that the GPU can provide a more flexible and cheap alternative to the use dedicated hardware in the NN field. The speedups attained, which are already impressive, are expected to increase even more, as new GPUs containing a greater number of cores arrive to the market. This allows for researchers and practitioners in pattern recognition to implement high quality NN solutions that could be disregarded otherwise, due to temporal and financial constraints. In future work online implementation of both algorithms will be considered.

References

1. Brandstetter, A., Artusi, A.: Radial basis function networks GPU based implementation. IEEE Transactions on Neural Networks 19(12), 2150–2154 (2008)
2. Vonk, E., Jain, L.C., Veelenturf, L.P.J.: Neural network applications. In: Jain, L.C. (ed.) Electronic Technology Directions, pp. 63–67. IEEE Computer Society, Los Alamitos (1995)
3. Catanzaro, B., Sundaram, N., Keutzer, K.: Fast support vector machine training and classification on graphics processors. In: Proc. of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, pp. 104–111 (2008)
4. Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J.W., Skadron, K.: A performance study of general-purpose applications on graphics processors using CUDA. Journal of Parallel and Distributed Computing 68(10), 1370–1380 (2008)
5. Steinkrau, D., Simard, P.Y., Buck, I.: Using GPUs for machine learning algorithms. In: Proc. 8th Int. Conf. on Doc, pp. 1115–1119. IEEE Computer Society, Los Alamitos (2005)
6. Jang, H., Park, A., Jung, K.: Neural network implementation using CUDA and OpenMP. In: DICTA 2008: Proc. of the 2008 Digital Image Computing: Techniques and Applications, Washington, DC, USA, pp. 155–161. IEEE Comp. Society, Los Alamitos (2008)
7. NVIDIA CUDA Programming Guide Version 2.2 (2009)
8. Lopes, N., Ribeiro, B.: An efficient gradient-based learning algorithm applied to neural networks with selective actuation neurons. Neural, Parallel & Scientific Computations 11(3), 253–272 (2003)
9. Lopes, N., Ribeiro, B.: Hybrid learning in a multi-neural network architecture. Neural Networks, 2001. In: Proc. of IJCNN 2001. Int Joint Conf. on Neural Networks, vol. 4, pp. 2788–2793 (2001)
10. WolframAlpha – computational knowledge engine, http://www.wolframalpha.com
11. Marques, A.: Feature extraction and PVC detection using neural networks and support vector machines. Master's thesis, University of Coimbra (2007)
12. Ribeiro, B., Marques, A., Henriques, J., Antunes, M.: Choosing real-time predictors for ventricular arrhythmia detection. IJPRAI 21(8), 1249–1263 (2007)

SPC without Control Limits and Normality Assumption: A New Method

J.A. Vazquez-Lopez and I. Lopez-Juarez*

Instituto Tecnologico de Celaya
Centro de Investigacion y de Estudios Avanzados del IPN - Unidad Saltillo
ismael.lopez@cinvestav.edu.mx

Abstract. Control Charts (CC) are important Statistic Process Control (SPC) tools developed in the 20's to control and improve the quality of industrial production. The use of CC requires visual inspection and human judgement to diagnoses the process quality properly. CC assume normal distribution in the observed variables to establish the control limits. However, this is a requirement difficult to meet in practice since skewness distributions are commonly observed. In this research, a novel method that neither requires control limits nor data normality is presented. The core of the method is based on the FuzzyARTMAP (FAM) Artificial Neural Network (ANN) that learns special and non-special patterns of variation and whose internal parameters are determined through experimental design to increase its efficiency. The proposed method was implemented successfully in a manufacturing process determining the statistical control state that validate our method.

Keywords: Control Charts, Neural Networks, Pattern Recognition.

1 Introduction

In manufacturing processes, the use of Control Charts (CC) is a common technique used to monitor the quality of the production. Variables are monitored to preserve the process under statistical control and also to detect any special variation. Should this situation occurs, then specially trained personnel take the appropriate actions to get the process back into control. By using CC it is possible to know when the process presents a special behaviour by monitoring its upper and lower control limits. However, using this approach, it is not possible to determine the type of pattern. To overcome this limitation, a novel method to recognise and analyse statistical quality patterns using the Fuzzy ARTMAP (FAM) Artificial Neural Network (ANN) is proposed. The FAM network parameters are determined off-line using experimental design and the Monte Carlo method which constitutes a novel method to increase the FAM efficiency eliminating the trial and error procedure commonly used [11]. During testing, the FAM Learning parameters are selected automatically depending if special or non-special pattern is encountered. The system is able to recognise both pattern types such as non special: natural in control; and special: upward shift,

* Corresponding author.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 611–618, 2009.

downward shift, upward trend and downward trend. In order to improve the discrimination new patterns can be added to the Initial Knowledge Base (IKB) forming what it is referred to as the Enhanced Knowledge Base (EKB). The network is retrained on-line to take into account the new pattern information which improves the pattern recognition capability of the system. The proposed method consists of two modules, the learning process and the control process. The first module includes basically the IKB and the EKB whereas the second module includes the pattern recognition and the control stages.

The use of this new method also begins a new Statistical Process Control (SPC) methodology since data normality is not required in the probability distribution of manufacturing processes as it was required in earlier production systems (not automated) using CC.

The rest of this article is organised as follows. In section 2, background information in terms of related work done by other researchers is presented while in section 3, the Fuzzy ARTMAP neural network is described in detail. Section 4 formally presents and describes the developed method. Results are given in section 5 and finally conclusions are provided and further work envisaged in section 6.

2 Related Work and Original Contribution

Considering the disadvantages of the CC, diverse investigations suggest the use of ANN as an alternative ([13], [4], [7], [3], [12], [14], [2], and [8]). The advantages of using ANN's in comparison with CC are: a) It is possible to work in real-time [5]. b) The assumption of data normality is not necessary [9]; and c), great amounts of complex data can be processed in a short time [10]. Hindi, used the Fuzzy ARTMAP to determine the type of change presented in the process parameters [6]. He compares the results with the obtained from the application of the $\bar{X}$ and R-chart. He used values 0 and 3 for μ and 1 and 3 for σ, considering the combination $\mu = 0$ and $\sigma = 1$ to represent a state of statistical control. The FAM parameter values were fixed. Guh, proposed the use of ANN Back-Propagation (BPN) in combination with a decision tree for pattern recognition [5]. In his work, Guh makes reference to three modules. Module A is in charge of data preprocessing, module B works like a CC detecting abnormal cases of variation whereas the module C determines the type of pattern based on a pre-defined decision tree. Our method compares favourably to previous work having the following advantages:

- Network parameters are determined through experimental design for maximum efficiency.
- Normality assumption is not required.
- High sampling size to guarantee data normality is not required.
- ANN testing parameters are selected automatically according to the type of probability distribution.
- The model refines its knowledge through real-world data.

3 FuzzyARTMAP (FAM)

The FuzzyARTMAP neural network is based on the Adaptive Resonance Theory (ART) which was developed by Stephen Grossberg and Gail Carpenter at Boston University. In Fuzzy ARTMAP there are two modules ART_a and ART_b and an inter-ART module "map field" that controls the learning of an associative map from ART_a recognition categories to ART_b recognition categories [1]. The FAM architecture is shown in figure 1.

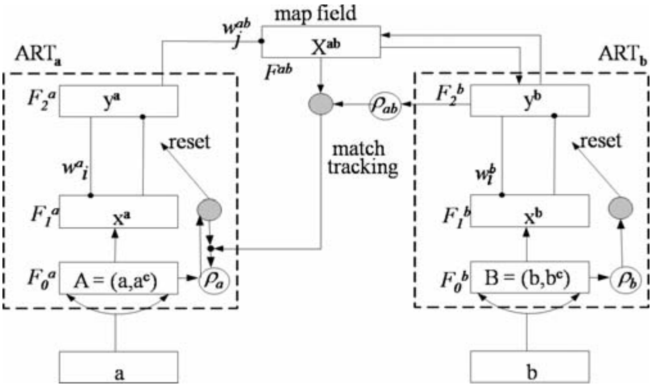


Fig. 1. FuzzyARTMAP Architecture

The map field module also controls the match tracking of ART_a vigilance parameter. A mismatch between Map field and ART_a category activated by input **a** and ART_b category activated by input **b** increases ART_a vigilance by the minimum amount needed for the system to search for, and if necessary, learn a new ART_a category whose prediction matches the ART_b category. The search initiated by the inter-ART reset can shift attention to a novel cluster of features that can be incorporated through learning into a new ART_a recognition category, which can then be linked to a new ART prediction via associative learning at the Map field. The algorithm uses a preprocessing step, called complement coding which is designed to avoid category proliferation. Similar to ART-1, a vigilance parameter measures the difference allowed between the input data and the stored pattern. Therefore this parameter is determinant to affect the selectivity or granularity of the network prediction. For learning, the FuzzyARTMAP has 4 important factors: Vigilance in the input module (ρ_a), vigilance in the output module (ρ_b), vigilance in the Map field (ρ_{ab}) and learning rate (β). These were the considered factors in this research. The FAM algorithm was coded in C++ using the Visual Studio 2005 compiler running in a Core2Duo PC computer at 1.86 GHz.

4 Method

The proposed method consists of three important elements: Learning process, Control process and the use of the EKB as shown in figure 2. The first element contains tree sub-elements; the random variable from groups of 20 data points[1] and the Fuzzy ARTMAP learning using either the IKB or the IKB + EKB.

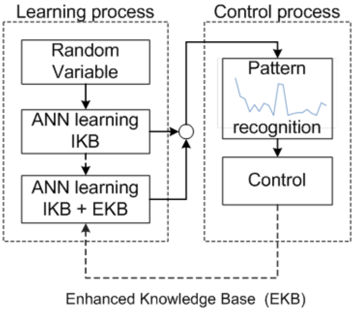


Fig. 2. General Method

Initially the FuzzyARTMAP network is trained with an IKB that was obtained from the Monte Carlo simulation. The data considered special and non special patterns as defined by the following equation: $X_t = \mu + n_t + d_t$ (1) where μ is the effect of the global data (mean), X_t, n_t and d_t are the the data, the effect of the natural variation and the effect of the special variation in time t, respectively [5]. A sample data with a natural pattern has $d_t = 0$, if it is unnatural, then $d_t > 0$. The Control process is formed by the pattern recognition and the control stages, which are better explained using figure 3. Finally, the EKB serves to add unknown patterns to the ANN learning.

4.1 Pattern Recognition

Figure 3 shows the algorithm to maintain the process under statistical control. The algorithm uses two stages, the pattern recognition and control. During pattern recognition the X vector is preprocessed as follows:

Standardization. The standardization of the X vector is obtained by $Y = (X - \varepsilon)/\tau$. where ε and τ are objective values for de mean and the standard deviation respectively. The real mean and standard deviation of X vector are $\bar{X}$ and S, respectively. If $\bar{X} \to \varepsilon$, and $S \to \tau$, then the X vector will be a non special pattern. Otherwise, the X vector will be a special pattern and the pattern has to be identified among the upward or downward shift, upward or downward trend or any other possible special pattern. The other special patterns can be cyclical, systematic or mixture. These other special patterns were not considered in this

[1] Using this group size an efficiency higher than 90% was obtained during experiments.

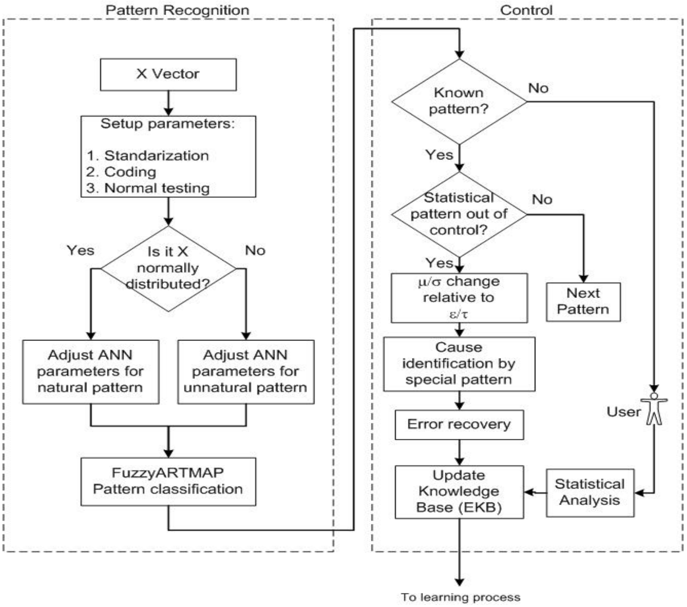

Pattern Recognition
Control
X Vector
Setup parameters:
1. Standarization
2. Coding
3. Normal testing
Is it X normally distributed?
Yes
No
Adjust ANN parameters for natural pattern
Adjust ANN parameters for unnatural pattern
FuzzyARTMAP Pattern classification
Known pattern?
No
Yes
Statistical pattern out of control?
No
Yes
μ/σ change relative to ε/τ
Next Pattern
Cause identification by special pattern
Error recovery
User
Update Knowledge Base (EKB)
Statistical Analysis
To learning process

Fig. 3. Process Control Method

paper since they are more dependent from the real process and its inclusion is considered in the EKB. All special pattern data cannot have only non normal distribution and this is the reason to group them in one IKB. Another IKB was integrated by only the X vector with normal distribution, but, is possible to find special and non special pattern data in it. For example, if $\bar{X}$ or S are near to ε and τ, but not sufficiently, the X vector can be special pattern data with normal distribution and natural pattern too. If a is the absolute difference between ε and $\bar{X}$ or between τ and S, then, there exists a numeric value given by a so that the probability distribution of X is not normal.

Coding. The coding stage consist of transforming the X vector within the range [0,1] using a lineal transformation.

Test for normal distribution This test is important since it indicates the type of FuzzyARTMAP learning parameters to be used depending if the data set is normally distributed or not. After selecting the proper ANN parameters the learning starts by training the FuzzyARTMAP network using the IKB as a training set which establishes the internal representation of the network.

4.2 Control

Once the statistical pattern is recognised, the control stage begins. If the pattern is not recognised by the FAM network, then user intervention is required to statistically analyze and classify the pattern. On the other hand the pattern can be classified as belonging to *process in control* or *process out of control.* A pattern belonging to a process in control indicates stability in the manufacturing process. In this case, a new X vector will be presented. Otherwise, the manufacture process is out control, then the special pattern is identified and a proper error recovery strategy implemented. When a special pattern occurs the first time, this pattern will be different from the simulated patterns used to train the FAM and it contains real data from the process so this information is used to enhance the previous knowledge base (IKB) forming the new EKB. In practice, this resulted in a *fine tuning* mechanism to increase the expertise and robustness of the neural network as demonstrated by the obtained results in a real process as indicated in the following section.

5 Issues and Experiments

The FuzzyARTMAP learning parameters considered for the experimental work are: the input base vigilance ($\rho_{a_{1,2}}$), vigilance in the map-field ($\rho_{ab_{1,2}}$), learning rate ($\beta_{1,2}$) and the output base vigilance ($\rho_{b_{1,2}}$)[2].

After trials with different parameters, a factorial fractional design (2^k_{III}), with 7 factors, 2 replicates, 16 experimental runs and a significance level of 0.05 resulted appropriate. The values for the factors levels were 0.2 and 0.8 for all them, with the exception of $\beta_{1,2}$ where the high level was set to 1.0. This experimentation employed the Monte Carlo Simulation method to generate the train and test set of vectors with special and non special patterns. X Vectors with non special natural pattern (a), special natural pattern (b), and special unnatural pattern (c) were considered during the experiments. Cases (a) and (b) have normal distribution and (c) non normal. For case (a), it was considered $\mu = 0$ and $\sigma = 1$; case (b) considered the combination (μ, σ) with values (0,3), (3,1) and (3,3). Finally, for the third case (c), the special patterns were obtained varying the term d in equation (1). The upward shift pattern was generated with $d > 2.5\sigma$, the downward shift was generated with $d < -2.5\sigma$. The upward trend pattern was generated with $d > 0.1\sigma$ and for the downward shift $d < -0.1\sigma$. A detailed explanation of the simulation and the selection of the network parameters it is given in our previous work [11]. The best parameters values determined after the experimental design are shown in the Table 1. These values are adjusted automatically during the training phase of the FuzzyARTMAP network depending on the type of detected pattern.

The validation of our method was carried out using simulated and real-world data[3]. The data presented in this paper comes from a make-up manufacturing company representing the level of quality conformance during the product packing stage. The process generated values which are the fraction of non-conform products (X vector). In the method, considering the objective non-conform fraction ($p = 0.005$) resulted in $\varepsilon = 0.69$ and $\tau = 0.83$. The number of X vectors processed by the algorithm are 21. The results are given in table 2.

[2] Subindexes 1,2 indicate parameters used during training/testing phase, respectively.

[3] Due to space limitation, only results from one industrial process are presented.

Table 1. FuzzyARTMAP parameters

ANN Parameter	*Normal Dist. Pattern*	*Nonnormal Dist. Pattern*
ρ_{a_1}	0.2	0.2
ρ_{a_2}	0.2	0.2
$\rho_{b_{1,2}}$	0.9	> 0.7
ρ_{ab_1}	any	0.2
ρ_{ab_2}	any	0.8
β_1	1.0	0.2
β_2	1.0	1.0

Table 2. Results form the packing process

Vector	*Distribution*	*Process*	*Results (CC)*	*Results (ANN)*
1	Non normal	Out control	Out control-downward trend	Downward trend
2 to 6	Non normal	Out control	In control	Downward trend
7	Non normal	Out control	Out control-downward trend	Downward trend
8	Non normal	Out control	In control	Downward trend
9	Non normal	Out control	Out control-downward trend	Downward trend
10,11	Non normal	Out control	In control	Downward trend
12 to 19	Normal	In control	In control	Natural In control
20	Non normal	Out control	Out control-downward trend	Downward trend
21	Non normal	Out control	Out control	Downward trend

6 Conclusions and Ongoing Work

An alternative method for Statistical Proces Control (SPC) that does not require control limits, assumption of normality in the process data and high simple size (to secure data normality) was presented.

It was observed that grouping family vectors according to its probability distribution (normal or non normal) and using two sets of network training parameters increased the neural network efficiency. The method was tested during simulation and also with process data from a make-up company comparing the obtained results with the use of Control Charts (CC). The proposed method compared favourably considering that it can be applied to either continuous or discrete variables, The efficiency of the method is improved when unknown patterns are added to the Initial Knowledge Base using the incremental learning properties of the FuzzyARTMAP network. The method is intended to be applied on-line to automatically monitor the quality of the production, therefore on going work is looking at the implementation in other industrial sectors.

Acknowledgments. The authors wish to thank CONACyT for the financial support given to the project through grant 61373 and also for the scholarship granted to J. A. Vazquez-Lopez during his doctoral studies.

References

1. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: FuzzyARTMAP: A neural network architecture for incremental learning of analog multidimensional maps. IEEE Trans. on Neural Networks 3(5), 698–713 (1992)
2. Cheng, C.S.: A neural network approach for the analysis of control chart patterns. International Journal of Production Research 35(3), 667–697 (1997)
3. Guh, R.S., Tannock, J.D.T.: Recognition of control chart concurrent patterns using a neural network approach. Int. J. Prod. Res. 37(8), 1743–1765 (1999)
4. Guh, R.S.: Robustness of the neural network based control chart pattern recognition system to non-normality. Int. J. Qual. Reliability Mgmt 19(1), 97–112 (2002)
5. Guh, R.S.: Real-time pattern recognition in statistical process control: a hybrid neural network/decision tree-based approach. IMechE. J. Engineering Manufacture 219, Part B (2005)
6. Al-Hindi, H.A.: Control Chart Interpretation Using Fuzzy ARTMAP. Journal of King Saud University. Engineering Sciences 16(1) (2004)
7. Ho, E.S., Chang, S.I.: An integrated neural network approach for simultaneous monitoring of process mean and variance shifts - a comparative study. Int. J. Prod. Res. 37(8), 1881–1901 (1999)
8. Hwarng, H.B., Chong, C.W.: Detecting process nonrandomness through a fast and cumulative learning ART-based pattern recognizer. Int. J. Prod. Res. 33(7), 1817–1833 (1995)
9. Pacella, M., Semeraro, Q., Anglania, A.: Manufacturing quality control by means of a Fuzzy ART network trained on natural process data. Engineering Applications of Artificial Intelligence 17, 83–96 (2004)
10. Pacella, M., Semeraro, Q.: Understanding ART-based neural algorithms as statistical tools for manufacturing process quality control. Engineering Applications of Artificial Intelligence 18, 645–662 (2005)
11. Vazquez-Lopez, J.A., Lopez-Juarez, I., Peña-Cabrera, M.: Experimental Design applied to the FuzzyARTMAP neural network to the statistical special pattern recognition. In: XVIII Congress of the Automatic Control Chilean Association-IFAC-IEEE (2008) (in spanish)
12. Wani, M.A., Pham, D.T.: Efficient control chart pattern recognition through synergistic and distributed artificial neural networks. Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture 213(B2), 157–169 (1999)
13. Zobel, C.W., Cook, D.F., Nottingham, Q.J.: An augmented neural network classification approach to detecting mean shifts in correlated manufacturing process parameters. International Journal of Production Research 42(4), 741–758 (2004)
14. Zorriassantine, F., Tannock, J.D.T.: A review of neural networks for statistical process control. Journal of Intelligent Manufacturing 9, 209–224 (1998)

XI Neural Networks for Pattern Recognition

Improved Online Support Vector Machines Spam Filtering Using String Kernels

Ola Amayri and Nizar Bouguila

Concordia University, Montreal,
Quebec, Canada H3G 2W1
{o_amayri,bouguila}@encs.concordia.ca

Abstract. A major bottleneck in electronic communications is the enormous dissemination of spam emails. Developing of suitable filters that can adequately capture those emails and achieve high performance rate become a main concern. Support vector machines (SVMs) have made a large contribution to the development of spam email filtering. Based on SVMs, the crucial problems in email classification are feature mapping of input emails and the choice of the kernels. In this paper, we present thorough investigation of several distance-based kernels and propose the use of string kernels and prove its efficiency in blocking spam emails. We detail a feature mapping variants in text classification (TC) that yield improved performance for the standard SVMs in filtering task. Furthermore, to cope for realtime scenarios we propose an online active framework for spam filtering.

Keywords: Support Vector Machines, Feature Mapping, Spam, Online Active, String Kernels.

1 Introduction

Electronic mail has gained immense usage in everyday communication for different purposes, due to its convenient, economical, fast and easy to use nature over traditional methods. Beyond the rapid proliferation of legitimate emails lies adaptive proliferation of unwanted emails that take the advantage of the internet, known as spam emails. Variety of techniques have been developed to mitigate sufferings of spam emails. In particular, many machine learning (ML) techniques have been employed in the sake of spam filtering such as Boosting Trees, k-nearest neighbor classifier, Rocchio algorithm, Naive Bayesian classifier, Ripper and SVMs [4]. SVMs have made a large contribution to the development of spam email filtering. Based on SVMs, different schemes have been proposed through TC approaches. Recent studies on spam filtering, using SVMs, have focused on deploying classical kernels which neglects the structure and the nature of the text. Along with common Bag-of-Word (BoW) feature mapping approach in a batch mode [6]. In this paper, we propose an automated spam filtering in realtime, that improves the blocking of spam emails and reduce the misclassification of legitimate emails. To reach this goal, we focus on three key aspects:

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 621–628, 2009.

first we explore several feature mapping strategies in context of text categorization. We intensively investigate the effect of various combinations of term frequency, importance weight and normalization on spam filtering performance. Second, we compare and analyze the use of various string kernels and different distance-based kernels for spam filtering. In addition, we provide detailed results for a fair comparison between different feature mapping and kernel classes using typical spam filtering criteria. Finally, we propose a framework of various online modes for spam filtering. We discuss the use of online SVMs, Transductive Support Vector Machines (TSVMs) and Active Online SVMs for spam filtering. We study proposed modes using different feature mapping and kernel classes, also.

This paper is organized as follows: in next section number of feature mapping choices that have been employed to transform email data into feature vectors usable by machine learning methods are outlined. In section 3 we briefly introduce Support Vector Machine, along with investigation of different kernels classes used in spam filtering tasks. Section 4 describes different online SVMs modes. In section 5 we report empirical results of proposed approaches. Finally conclusions are presented in Section 6.

2 Input Data Format

Many researchers have pointed out the importance of text representation in the performance of TC using SVMs. In this section, briefly, we discuss different approaches that have been applied in text representation. Generally, supervised TC is engaged into three main phases: term selection, term weighting, and classifier learning. Among existing approaches, the text representation dissimilarity can be shown either on what one regards the meaningful units of text or what approach one seeks to compute term weight. Terms are usually identified with words syntactically or statistically. In BoW, for instance, the extraction of features is based on defining a substring of contiguous characters *word* w, where word boundary is specified using a set of symbolic delimiters such as whitespace, etc. Using k-mer (i.e. k-gram)approach, however, the document can be represented by predefined sequences of contiguous characters (i.e. sub-strings) of length k. Moreover, term weighting phase is a vital step in TC, involves converting each document d to vector space which can be efficiently processed by SVMs. Term weights can be considered by occurrence of term in the corpus (term frequency) or by its presence or absence (binary). In our experiments we adopted seven term weighting schemes similar to [8]. In particular, the first three term weighting schemes are different variants of term frequency which are: TF, $\log TF$ and ITF. Next four schemes are different combinations of term frequency and importance weight which are: $TF\text{-}IDF$, $\log TF\text{-}IDF$ and $ITF\text{-}IDF$. For large corpus, if we consider each distinct feature for spam filtering then a very dense feature space F is constructed. To solve this issue researchers suggest *Stop words* and *Stemming*. More sophisticated feature selection is found by computing the probability of dependency between term w and category c such as Information Gain (IG), CHI statistic (χ^2) and Term strength (TS). In addition, spammers attempt to defeat the spam filtering by writing short emails. To handle such problem, in our experiments, we normalize emails by using L_2-normalization which yields generally to best error bounds.

3 Support Vector Machines: Kernels

SVMs are known to give accurate discrimination in high feature space [3]. Furthermore, they received a great attention in many applications such as text classification. The state of the art of SVMs evolved mapping the learning data from input space into higher dimensional feature space where the classification performance is increased. This has been developed by applying several kernels each with individual characteristics. Lately, the choice of the kernel became a widely discussed issue, since it reveals different performance result for various applications. SVMs in classification problems, such as spam filtering, explore the similarity between input emails implicitly using inner product $K(X, Y) = \langle\phi(X), \phi(Y)\rangle$ i.e. kernel functions. In distance based learning [12] the data samples $\overrightarrow{x}$ are not given explicitly but only by a distance function $d(\overrightarrow{x}, \overrightarrow{x}')$. In our experiments we compare the effectiveness of different kernels in this class which are: *Gaussian*, *Laplacian*, χ^2, *Inv multi*, *Polynomial*, and *Sigmoid* [12]. In contrast of distance-based kernels, string kernels define the similarity between pair of documents by measuring the total occurrence of shared substrings of length k in feature space F. In this case, the kernel is defined via an explicit feature map. In our experiments we adopted two classes of string kernels: the position-aware string kernel which takes advantage of positional information of characters/substrings in their parent strings and the position-unaware string kernel which does not. We applied Weighted Degree kernel (WD) and Weighted Degree kernel with Shift (WDs) [11] for position-aware kernels. Additionally, for position-unaware kernels, Subsequence String kernel (SSK) [10], Spectrum kernel and Inexact String Kernels such as Mismatch kernel, Wildcard kernel and Gappy kernel [9].

4 Support Vector Machines: Learning and Classification

In reality, spam filtering is typically tested and deployed in an online setting, by proceeding incrementally. To this end, Online SVM model presents to the filter a sequence of emails, where sequence order is determined by the design (i.e. it might be in chronological order or even randomized). We adopted a simple algorithm introduced in [7] to adapt batch model to online model. Initially, suppose a spam filter is trained on training set. In SVM model, examples closer to the hyperplane are most uncertain and informative. Those examples are presented by support vectors (SVs). Furthermore, SVs are able to summarize the data space and preserve the essential class boundary. Consequently, in our model, we use SVs as seeds (starting point) for the future retraining and discard all non-SVs samples. To this end, labeled data sets are not often affordable prior classification and label data set is time consuming and tedious process. To overcome this problem, TSVM constructs a maximum margin by employing large collection of unlabeled data jointly with a few labeled examples for improving generalization performance [5]. To cope with realtime scenario, Online Active SVM presents messages to the filter in a stream, where the filter must classify them one by one. Each time a new example is presented to the filter, the filter has the option of requesting a label for the given message using *Angle diversity* approach [1].

5 Experimental Results

Recently, spam filtering using SVM classifier has been tested and deployed using linear kernel weighted using binary weighting schemes [13,2,4]. We extend previous research on spam filtering, as we consider three main tasks. Firstly, we compare the use of various feature mapping techniques described in section 2 for spam email filtering. Secondly, we investigate the use of string kernels with a number of classical kernels and exploring that in terms of accuracy, precision, recall, F1 and running classification time. Thirdly, we report results from experiments testing the effectiveness of the online, TSVM and online active learning methods, presented in previous sections. In seek of comparison, the performance of each task is examined using the same version of spam data set which is trec05-p1[1] (92,189 labeled spam and legitimate emails) and the same pre-processing is applied for different kernels. In the purpose of comparison evaluation, SVM^{light}[2] package was used as an implementation of SVMs. We set the value of ρ in

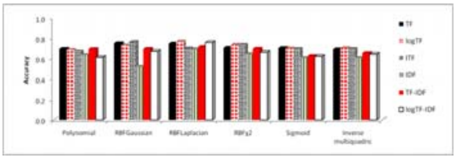


Fig. 1. The performance of SVM spam filtering on trec05-1, where IG has been applied

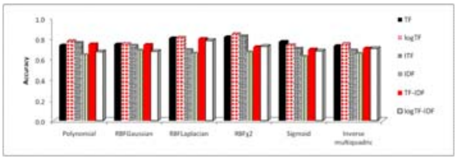


Fig. 2. The performance of SVM spam filtering on trec05-1, where χ^2 has been applied

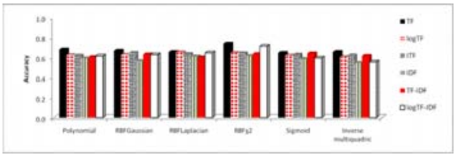


Fig. 3. The performance of SVM spam filtering on trec05-1, where TS has been applied

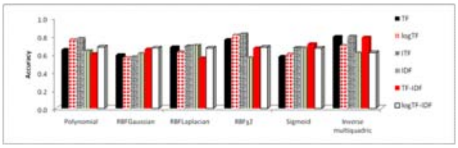


Fig. 4. The performance of SVM spam filtering on trec05-1, where stop words list and stemming have been applied

[1] http://plg1.cs.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo

[2] http://svmlight.joachims.org/

Table 1. The performance of batch SVM (SVM), TSVM, Online SVM (ON) and Online Active SVM (ONA) spam filtering on trec05-1 using distance-based kernels normalized using L_2-norm, and without removing stop words

Kernel	Precision				Recall				F1			
	SVM	TSVM	ON	ONA	SVM	TSVM	ON	ONA	SVM	TSVM	ON	ONA
Polynomial.TF	0.779	0.800	0.798	0.787	0.805	0.807	0.810	0.817	0.792	0.804	0.804	0.802
Gaussian.TF	0.876	0.824	0.870	0.869	0.850	0.804	0.858	0.842	0.863	0.814	0.864	0.855
Laplacian.TF	0.919	0.903	0.904	0.919	0.879	0.843	0.868	0.863	0.899	0.872	0.886	0.890
χ^2.TF	0.893	0.892	0.891	0.898	0.879	0.876	0.880	0.844	0.886	0.884	0.886	0.870
Sigmoid.TF	0.897	0.897	0.876	0.899	0.817	0.788	0.820	0.794	0.855	0.839	0.847	0.843
Inv multi.TF	0.779	0.791	0.798	0.787	0.824	0.807	0.868	0.813	0.801	0.799	0.831	0.799
Polynomial.logTF	0.854	0.858	0.850	0.861	0.838	0.836	0.842	0.823	0.846	0.847	0.846	0.842
Gaussian.logTF	0.819	0.815	0.833	0.837	0.849	0.824	0.853	0.835	0.834	0.820	0.843	0.836
Laplacian.logTF	0.901	0.871	**0.910**	0.885	0.893	0.887	0.890	0.888	0.897	0.879	0.900	0.886
χ^2.logTF	**0.920**	**0.918**	0.903	**0.921**	**0.915**	**0.908**	**0.916**	**0.906**	**0.917**	**0.913**	**0.910**	**0.913**
Sigmoid.logTF	0.803	0.803	0.815	0.795	0.836	0.812	0.840	0.825	0.819	0.808	0.827	0.809
Inv multi.logTF	0.797	0.799	0.789	0.765	0.873	0.841	0.882	0.859	0.833	0.820	0.833	0.809
Polynomial.ITF	0.835	0.857	0.831	0.827	0.836	0.803	0.845	0.818	0.835	0.829	0.838	0.822
Gaussian.ITF	0.872	0.812	0.889	0.863	0.796	0.792	0.801	0.790	0.832	0.802	0.843	0.825
Laplacian.ITF	0.793	0.749	0.836	0.765	0.790	0.772	0.790	0.772	0.792	0.760	0.812	0.768
χ^2.ITF	0.899	0.892	0.905	0.897	0.893	0.892	0.883	0.887	0.896	0.892	0.894	0.892
Sigmoid.ITF	0.769	0.783	0.797	0.774	0.795	0.765	0.783	0.793	0.782	0.774	0.790	0.783
Inv multi.ITF	0.795	0.742	0.800	0.778	0.788	0.769	0.795	0.760	0.791	0.756	0.797	0.769
Polynomial.IDF	0.683	0.708	0.698	0.679	0.768	0.716	0.778	0.757	0.723	0.712	0.736	0.716
Gaussian.IDF	0.778	0.743	0.759	0.758	0.765	0.769	0.780	0.785	0.771	0.756	0.769	0.771
Laplacian.IDF	0.747	0.746	0.760	0.729	0.801	0.709	0.812	0.786	0.773	0.727	0.785	0.756
χ^2.IDF	0.719	0.713	0.729	0.732	0.737	0.767	0.759	0.778	0.728	0.739	0.744	0.755
Sigmoid.IDF	0.748	0.703	0.799	0.735	0.709	0.689	0.714	0.696	0.728	0.696	0.754	0.715
Inv multi.IDF	0.728	0.749	0.718	0.737	0.698	0.701	0.733	0.674	0.713	0.725	0.725	0.704
Polynomial.TF-IDF	0.854	0.807	0.850	0.836	0.823	0.825	0.835	0.809	0.838	0.816	0.842	0.822
Gaussian.TF -IDF	0.836	0.802	0.842	0.831	0.835	0.817	0.846	0.826	0.835	0.809	0.844	0.829
Laplacian.TF-IDF	0.832	0.850	0.820	0.861	0.889	0.885	0.898	0.893	0.860	0.867	0.857	0.877
χ^2.TF-IDF	0.816	0.816	0.818	0.808	0.793	0.761	0.796	0.785	0.804	0.788	0.807	0.796
Sigmoid.TF-IDF	0.797	0.749	0.790	0.794	0.831	0.782	0.842	0.809	0.814	0.765	0.815	0.801
Inv multi.TF-IDF	0.796	0.749	0.790	0.788	0.811	0.804	0.820	0.816	0.803	0.776	0.805	0.801
Polynomial.logTF-IDF	0.742	0.747	0.768	0.732	0.752	0.740	0.769	0.745	0.747	0.743	0.768	0.739
Gaussian.logTF-IDF	0.726	0.793	0.758	0.739	0.769	0.705	0.773	0.747	0.747	0.747	0.766	0.743
Laplacian.logTF-IDF	0.827	0.840	0.831	0.818	0.866	0.863	0.866	0.888	0.846	0.851	0.848	0.851
χ^2.logTF-IDF	0.817	0.803	0.820	0.800	0.891	0.786	0.892	0.793	0.852	0.795	0.855	0.797
Sigmoid.logTF-IDF	0.813	0.790	0.832	0.810	0.795	0.712	0.787	0.767	0.804	0.749	0.809	0.788
Inv multi.logTF-IDF	0.789	0.824	0.779	0.799	0.775	0.732	0.786	0.766	0.782	0.776	0.782	0.782

RBF kernels, and C for the soft margin via 10-fold cross-validation. For TSVM, the value of C^* is set similar to C. We ran experiments for similar length of substrings used in string kernels (value of k parameter). We varied the value of the decay vector λ for SSK to see its influence in the performance, where the higher value of λ gives more weight to non-contiguous substrings ($\lambda = 0.4$ has provided a better performance). For mismatch kernel (k, m), wildcard kernel (k,w) and gappy kernel (g,k), the experiments have taken place with fixed values of allowed mismatch, wild card and gaps which are $m = 1$, $w = 2$, $k = 2$,

Table 2. The performance of batch SVM, TSVM, Online SVM (ON) and Online Active SVM (ONA)spam filtering on trec05-1 using string kernels

Kernel	Precision				Recall				F1			
	TSVM	SVM	ONA	ON	TSVM	SVM	ONA	ON	TSVM	SVM	ONA	ON
SSk	0.9278	0.9509	0.9505	0.9590	0.9281	0.9531	0.9468	0.9729	0.9279	0.9404	0.9341	0.9372
Spectrum	0.9249	0.9657	0.9466	0.9800	0.9362	0.9345	0.9478	0.9479	0.9305	0.9325	0.9315	0.9320
Mismatch	0.8745	0.8967	0.9012	0.9033	0.9100	0.9277	0.9145	0.9265	0.8919	0.9094	0.9006	0.9050
Wildcard	0.9356	0.9689	0.9432	0.9900	0.8890	0.9056	0.8952	0.9112	0.9117	0.9086	0.9102	0.9094
Gappy	0.9190	0.9678	0.9256	0.9943	0.9167	0.9012	0.9189	0.9043	0.9178	0.9094	0.9136	0.9115
WD	0.8978	0.9571	0.9189	0.9700	0.9021	0.9100	0.9067	0.9190	0.8999	0.9049	0.9024	0.9037
WDs	0.8987	0.9124	0.9109	0.9167	0.9189	0.9080	0.9001	0.9088	0.9087	0.9083	0.9085	0.9084

Table 3. Execution time for different combinations of frequency and distance kernels in different modes: batch SVM (SVM) and online SVM (ON). These times do not include time spent in feature mapping.

	Polynomail		Gaussian		Laplacian		χ^2		Sigmoid		Inv multi	
	SVM	ON	SVM	ON	SVM	ON	SVM	ON	SVM	ON	SVM	ON
TF	0.03	0.03	0.4	0.3	3.48	2.4	0.49	0.38	0.1	0.1	3.43	3.1
logTF	0.45	0.4	1.25	1.1	5.54	4.44	5.47	5	6.36	6	3.21	2.45
ITF	1.55	1.35	2.39	1.55	**6.48**	**6.30**	5.49	4.55	3.03	3.03	4.02	4
IDF	0.47	0.46	2.02	2	2.05	2	4.57	4.5	3.07	3.01	4.11	4.05
TF-IDF	0.02	0.02	0.39	0.37	3	3	0.35	0.34	0.55	0.53	3.43	3.1
logTF-IDF	0.45	0.4	1.2	1.1	4.3	3.25	5	5	6.4	5.59	3.2	2.55

Table 4. Execution time for string kernels in different modes: batch SVM (SVM) and online SVM (ON). These times do not include time spent in feature mapping.

	SSk	Spectrum	Mismatch	Wildcard	Gappy	WD	WDs
SVM	20.08	19.45	**21.43**	20.47	20.02	19.56	20.32
ON	7.37	18.55	19.30	20.20	**19.32**	19.10	20.00

respectively, as allowing higher mismatch will increase the computation cost. To examine the use of different feature mapping, we evaluate trec05p-1 data set using generic combinations of feature mapping approaches along with different distance-based kernels (i.e. $Polynomial.TF$ all normalized using L_2). Clearly, classification performance is better when no feature selection techniques were applied. Figures 1, 2, 3 and 4 show slight degradation in performance comparing with the performance of spam filtering using all distinct words. Tables 1 through 2 show the comparison results obtained for distance-based and string kernels along with different feature mapping combinations deployed in different SVMs modes. We achieved best performance with emails weighted using different variant of TF and normalized using $L2$-norm. Besides, the kernel choice is crucial in classification problem. The good kernel is the kernel that gives a valuable information about the nature of data, and report good performance. RBF kernels have the higher performance among distance based kernels in most of experiments. For instance, string kernels, in particular SSK, yields improved performance compared to batch supervised learning, with reduced number of labels and reasonable computational time. On the basis of kernels comparison string kernels performed better than distance-based kernels. Besides F1, precision, and recall, we evaluate involved kernels in terms of their computational efficiency, in order to provide insight into the kernels impact on filtering time. We measured the duration of computation for all kernels (see results in Tables 3 and 4). As expected, string kernels were defeated by their computational cost [9]. In addition, results show a clear dominance of online active learning methods, compared to both Online SVM and TSVM.

6 Conclusion

The ultimate goal of our extensive study of automated spam filtering using SVMs is to develop a devoted filter for spam problem in order to improve the blocking rate of spam emails (high precision) and reduce the misclassification rate of legitimate emails (high recall). The path towards such powerful filter is a thorough study of powerful classifier to accurately distinguish spam emails from legitimate emails and to consider the dynamic nature of spam problem. In this paper, particularly, we intensively study SVM email classification performance given by deployed kernels in realtime environment. Indeed, we described the use of string kernels in order to improve spam filter performance. We implemented, tested, integrated various preprocessing algorithms based on term frequency, importance weight with normalization to investigate their impact on classifier performance. Moreover, we applied algorithms to adapt batch theoretical models to online real world models using string kernels and well-performed preprocessing combinations, and hence maximize the overall performance. Further enhancement can be made by taking into account user feedback and the structure of emails which is richer than only text.

References

1. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 59–66 (2003)
2. Cormack, G.V., Bratko, A.: Batch and on-line spam filter comparison. In: Proceedings of the Third Conference on Email and Anti-Spam, California, USA (2006)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(1), 229–273 (1995)
4. Drucker, H., Vapnik, V., Wu, D.: Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10(5), 1048–1054 (1999)
5. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the sixteenth International Conference on Machine Learning (ICML 1999), San Francisco, US, pp. 200–209 (1999)

6. Kolcz, A., Alspector, J.: Svm-based filtering of e-mail spam with content-specific misclassification costs. In: Proceedings of the Workshop on Text Mining, California, USA, pp. 123–130 (2001)
7. Lau, K.W., Wu, Q.H.: Online training of support vector machine. Pattern Recognition 36(8), 1913–1920 (2003)
8. Leopold, E., Kindermann, J.: Text categorization with support vector machines. how to represent texts in input space? Machine Learning 46(13), 423–444 (2002)
9. Leslie, C., Kuang, R.: Fast string kernels using inexact matching for protein sequences. Journal of Machine Learning Research 5, 1435–1455 (2004)
10. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. The Journal of Machine Learning Research 2(1), 419–444 (2002)
11. Rtsch, G., Sonnenburg, S., Schlkopf, B.: Rase: Recognition of alternatively spliced exons in c. elegans. Bioinformatics 21(1), i369–i377 (2005)
12. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
13. Sculley, D., Wachman, G.: Relaxed online svms for spam filtering. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, Amsterdam, Netherlands, pp. 415–422 (2007)

Neural Network Ensembles from Training Set Expansions

Debrup Chakraborty

Computer Science Department, CINVESTAV-IPN, Av. IPN No. 2508, Col. San
Pedro Zacatenco, Mexico, D.F. 07360, Mexico
debrup@cs.cinvestav.mx

Abstract. In this work we propose a new method to create neural network ensembles. Our methodology develops over the conventional technique of *bagging*, where multiple classifiers are trained using a single training data set by generating multiple bootstrap samples from the training data. We propose a new method of sampling using the k-nearest neighbor density estimates. Our sampling technique gives rise to more variability in the data sets than by bagging. We validate our method by testing on several real data sets and show that our method outperforms bagging.

1 Introduction

The goal of constructing an ensemble of classifiers is to train a diverse set of classifiers from a single available training data set, and to combine their outputs using a suitable aggregation function. In the past few years there have been numerous proposals for creating ensembles of classifiers, and in general, it have been noticed that an ensemble of classifiers have better generalization abilities than a single classifier. Two of the well known proposals for creating classifier ensembles are *bagging* [2] and *boosting* [12]. Ample theoretical and experimental studies of Bagging, Boosting and their variants have been reported in the literature, and these studies clearly point out why and under which scenarios ensembles created by these methods can give better predictions [2,9,10,13].

Bagging is a popular ensemble method which can significantly improve generalization abilities of "unstable" classifiers [2]. In bagging, given a training data set $\mathcal{L}_x = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \subset \Re^n$ with the associated class labels, α independent bootstrap samples [5] are drawn from $\mathcal{L}_x$ each of size m. In other words, from the original training set $\mathcal{L}_x$, α different sets $\mathcal{B}_1, \mathcal{B}_1, \ldots, \mathcal{B}_\alpha$ are obtained each containing m points with their associated labels. These α different sets thus obtained are used to train α different classifiers. In the discussions that follow we shall call a single member of the ensemble as a *candidate*. The final decision is made by an aggregation of the outputs of the candidates. The type of aggregation depends on the type of the output, i.e., whether it is a numerical response or a class label. Generally, for classification a majority voting type aggregation is applied, whereas in case of regression (function approximation) type problems

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 629–636, 2009.

an average or a weighted average is used. This simple procedure can decrease the classification error and give better classifiers with good generalization abilities. The intuitive reason of why bagging works is that each candidate learns a slightly different decision boundary, and thus the combination of all the different decision boundaries learned by the candidate classifiers give rise to less variance in the classification error. In [2] Leo Breiman provided theoretical justification of the fact that one can obtain significant improvement in performance by bagging unstable classifiers. It was also noted in [2] that supervised feed-forward neural networks like the multilayer perceptron (MLP) are unstable, i.e., it is not necessary that for a trained MLP, small changes in the input will produce small changes in the output. Thus it is expected that bagging can decrease classification errors in MLP classifiers to a large extent

Right from the early nineties neural network ensembles has been widely studied [8,14]. A class of studies regarding neural network ensembles are directed towards adapting suitably the general ensemble techniques in case of neural networks [4]. Other studies have been focussed on developing heuristics to choose better candidates for an ensemble such that each candidate has good prediction power along with that the selected candidates have better diversity [3,6], which is known to affect the performance of an ensemble [9,10].

In this paper we propose a new method to create neural network ensembles based on bagging. As discussed earlier, in bagging a bootstrap sample of a given training set is used to train a candidate classifier. A bootstrap sample is generated by sampling with replacement, so the difference among the various bootstrap samples is that there may be some data points missing or some data points may get repeated. In the proposed method we aim to achieve more diversity in each of the training set which would be used to train the candidates of the ensemble. In the ideal scenario it can be assumed that the training data gets generated from a fixed but unknown time-invariant probability distribution. It would have been the best if the different training sets for the candidates could have been independently generated following the same probability distribution from which the training data was generated. But, as this distribution is unknown, so such a method cannot be developed in practice. One of the closely related options can be to estimate the probability distribution of the training data and thus draw different training sets from this estimated distribution. Our work is motivated by this approach. The problem of this approach is that generally the number of available training data is too small to have a reasonable estimate of the distribution. So, in this work we do not attempt to estimate the true probability distribution of the training set, but we propose a method to generate new data points such that the new points are generated according to the spatial density of the training set, i.e., more points are generated in the dense regions of the data and less points in the sparse regions.

The heart of our method is the k-nearest neighbor (k-NN) density estimation and classification procedure. The new data points that are generated for training the candidates in a sense follows the k-NN density estimate of the original training data. This technique has been successfully used for data condensation in [11]. But we use it for a completely different goal. We generate new points for each candidate and mix these new points with the original training data and train the candidate with this data. Thus, it is expected that the training sets used for the candidates are more diverse than the bootstrap samples. Our experiments demonstrate that this technique when applied to MLP ensembles can give better results than conventional bagging.

2 $\boldsymbol{k}$ Nearest Neighbor Density Estimation

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \Re^n$ be independently generated from a continuous probability distribution with density f. The nearest neighbor density estimation procedure finds the density of a point $\boldsymbol{z}$. We describe the methodology in brief next.

Let $||\boldsymbol{x} - \boldsymbol{z}||$ denote the Euclidian distance between points $\boldsymbol{x}$ and $\boldsymbol{z}$. A n dimensional hyper-sphere centered at $\boldsymbol{x}$ with radius r is given by the set $S_{\boldsymbol{x},r} = \{\boldsymbol{z} \in \Re^n : ||\boldsymbol{x} - \boldsymbol{z}|| \leq r\}$. We call the volume of this sphere as $V_r = \mathsf{Vol}(S_{\boldsymbol{x},r})$. Let $k(N)$ be a sequence of positive integers such that $\lim_{N\to\infty} k(N) = \infty$ and $\lim_{N\to\infty} k(N)/N = 0$. Suppose we have a sample $\mathcal{L}_x = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \subset \Re^n$, and we fix a value of $k(N)$. Let $r_{k(N),\boldsymbol{z}}$ be the Euclidian distance of $\boldsymbol{z}$ from its $(k(N)+1)$-th nearest neighbor in $\mathcal{L}_x$. Then the density at $\boldsymbol{z}$ is estimated as

$$\hat{f}(\boldsymbol{z}) = \frac{k(N)}{N} \times \frac{1}{V_{r_{k(N),\boldsymbol{z}}}} \tag{1}$$

It has been shown that this estimate is asymptotically un-biased and consistent, but it is known that this estimate suffers from the curse of dimensionality, i.e., the estimate gets unstable for high dimensional data. We shall use this density estimation technique to generate new training points, which we describe next.

3 Expanding a Training Set

Our basic motivation is to increase the variability of the individual training sets which we shall use to train each candidate classifier. The idea is to create new training points which are similar to the ones in the original training set. Ideally, we want to generate points from the same probability distribution from which the training data was generated. As that distribution is unknown to us and obtaining a reasonable estimate from a small training set is not feasible we shall apply some heuristic to generate new points following the rule that more points should be generated in the denser regions of the distribution.

Given a labeled data set $\mathcal{L} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : \boldsymbol{x}_i \in \Re^n, \boldsymbol{y}_i \in \{1, 2, \ldots, c\}, i = 1, \ldots m\}$, we shall call the set of the input vectors as $\mathcal{L}_x = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$. We shall denote the label (or output) associated with $\boldsymbol{x}$ as $\ell(\boldsymbol{x})$. For each $\boldsymbol{x}_i$ we compute the distance of its k-th nearest neighbor in $\mathcal{L}_x$. We call this distance as d_i. From eq. (1), it is clear that the density at a point $\boldsymbol{x}_i$ is inversely related to the volume of the hypersphere centered at $\boldsymbol{x}_i$ with radius d_i. So, it can be inferred that points with higher values of d_i lies in less dense areas and the points which low values of d_i lies in denser areas. Our objective is to generate new points following the density of the original data, i.e., our method must be such that more points are generated in the denser regions and less points in sparse regions. To achieve this, for each $\boldsymbol{x}_i \in \mathcal{L}_x$ we define a quantity p as follows:

$$p(\boldsymbol{x}_i) = \frac{1}{Z} e^{-d_i} \tag{2}$$

where

$$Z = \sum_{i=1}^{m} e^{-d_i} \tag{3}$$

This definition of p guarantees that for a point $\boldsymbol{x}_i$ the value of $p(\boldsymbol{x}_i)$ would be large if d_i is small and vice versa. Also, because of the way we define p it is obvious that for all $\boldsymbol{x}_i \in \mathcal{L}_x$, $0 \leq p(\boldsymbol{x}_i) \leq 1$, and also $\sum_{i=1}^{m} p(\boldsymbol{x}_i) = 1$. Thus p can be treated as a discrete probability distribution on the set $\mathcal{L}_x$. To generate a single new point we first sample a point randomly from $\mathcal{L}_x$ according to the probability distribution p. The roullet wheel selection technique can be used for this purpose. Let $\boldsymbol{x} \in \mathcal{L}_x$ be the sampled point. As $\boldsymbol{x}$ has been sampled according to the probability p, with high probability it will lie in a dense region of the training data. Let $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\}$ be the k nearest neighbors of the sampled point $\boldsymbol{x}$. Let $\mathsf{NBRS}(\boldsymbol{x})$ be the set containing the k nearest neighbors of $\boldsymbol{x}$ along with $\boldsymbol{x}$, i.e,

$$\mathsf{NBRS}(\boldsymbol{x}) = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\} \cup \{\boldsymbol{x}\}.$$

We now generate the new point $\tilde{\boldsymbol{x}}$ as a random convex combination of the points in $\mathsf{NBRS}(\boldsymbol{x})$. In other words, let each λ_j, for $j = 1, \ldots, k+1$, be generated independently from a uniform random distribution over $[0, 1]$, we compute $\tilde{\boldsymbol{x}}$ as

$$\tilde{\boldsymbol{x}} = \frac{\sum_{j=1}^{k} \lambda_j \boldsymbol{z}_j + \lambda_{k+1}\boldsymbol{x}}{\sum_{j=1}^{k+1} \lambda_j}. \tag{4}$$

The new point will thus lie within the convex hull of the points in $\mathsf{NBRS}(\boldsymbol{x})$, and thus cannot be very atypical of the points already present in the training set.

The new point $\tilde{\boldsymbol{x}}$ was not present in the training set, so to incorporate it into the training set we need to label this point, i.e., assign a target output to this point. The most natural label of $\tilde{\boldsymbol{x}}$ would be that label which the majority of its neighbors have. Note, that $\mathsf{NBRS}(\boldsymbol{x})$ are the $k+1$ neighbors of $\tilde{\boldsymbol{x}}$ (including $\boldsymbol{x}$ itself). Thus, the label of $\tilde{\boldsymbol{x}}$ is calculated as

$$\ell(\tilde{\boldsymbol{x}}) = \underset{j=1,\ldots,c}{\operatorname{argmax}} \sum_{\boldsymbol{z} \in \mathsf{NBRS}(\boldsymbol{x})} \delta(j, \ell(\boldsymbol{z})),$$

where $\delta(a, b) = 1$ if $a = b$, and $\delta(a, b) = 0$ if $a \neq b$.

The method described above can be repeated to obtain the desired number of new points. The algorithm in Fig. 1 summarizes the procedure described above. The algorithm Expand as described in Fig. 1 takes as input the training set $\mathcal{L}$, along with the parameters k and ν, where ν is the number of points that

```
Algorithm Expand($\mathcal{L}$,$k$,$\nu$)
    $Z \leftarrow 0$;
    for $i = 1$ to $m$;
        $d_i \leftarrow$ Distance of the $k$-th nearest neighbor of $\boldsymbol{x}_i$ in $\mathcal{L}_x$;
        $p(\boldsymbol{x}_i) \leftarrow e^{-d_i}$;
        $Z \leftarrow Z + p(\boldsymbol{x}_i)$
    end for
    for $i = 1$ to $m$,
        $p(\boldsymbol{x}_i) \leftarrow p(\boldsymbol{x}_i)/Z$;
    end for
    NewPoints $\leftarrow \emptyset$;
    while $|$NewPoints$| < \nu$,
        Select $\boldsymbol{x}$ from $\mathcal{L}_x$ with probability $p(\boldsymbol{x})$
        $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\} \leftarrow k$ nearest neighbors of $\boldsymbol{x}$ in $\mathcal{L}_x$;
        /* Let NBRS($\boldsymbol{x}$) $= \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\} \cup \boldsymbol{x}$ */
        $\lambda_1, \lambda_2, \ldots, \lambda_{k+1} \sim U[0,1]$; $\Lambda \leftarrow \sum_{i=1}^{k+1} \lambda_i$
        $\tilde{\boldsymbol{x}} \leftarrow (\sum_{i=1}^{k} \lambda_i \boldsymbol{z}_i + \lambda_{k+1}\boldsymbol{x})/\Lambda$;
        $\ell(\tilde{\boldsymbol{x}}) \leftarrow \mathrm{argmax}_{j=1,\ldots,c} \sum_{\boldsymbol{z} \in \mathsf{NBRS}(\boldsymbol{x})} \delta(j, \ell(\boldsymbol{z}))$ ;
        NewPoints $\leftarrow$ NewPoints $\cup \{(\tilde{\boldsymbol{x}}, \ell(\tilde{\boldsymbol{x}}))\}$
    end while
    return NewPoints;
```

Fig. 1. Algorithm to expand a training set

are required to be generated. It gives as output a set called NewPoints, which contains ν many new points generated by the procedure.

4 Creating the Ensemble

Our strategy of creating the ensemble closely follows bagging, except the fact that instead of using bootstrap samples for training the candidates of the ensemble we use the algorithm Expand of Fig. 1 to create new points and mix them with the original training set. Given a training set $\mathcal{L}$ we decide upon the size of the ensemble, i.e., the number of candidate classifiers. Let us call this as α. We fix two integers k and ν and call Expand$(\mathcal{L}, k, \nu)$ α times. By this way we obtain $S_1, S_2, \ldots, S_\alpha$ as output, where each S_i contains ν points. For training the i-th candidate we train a multilayered perceptron using $\mathcal{L} \cup S_i$. Thus obtaining α trained networks. For using the network, we feed a test point to all the α networks and decide the class of the test point by a majority vote. The algorithm for creating the ensemble is depicted in Fig. 2.

The algorithm Create_Ensemble takes as input the training set $\mathcal{L}$, k, the number of new points to be used for each candidate ν and the size of the ensemble α. The algorithm calls a function Train, which takes as input a training set and a variable $\mathcal{A}$ which contains the parameters necessary to fix the architecture of a network. The algorithm Train outputs a vector $\boldsymbol{W}$ which contains the weights

```
Algorithm Create_Ensemble($\mathcal{L}$,$k$,$\nu$,$\alpha$)
    for $i = 1$ to $\alpha$;
        $S_i \leftarrow$ Expand($\mathcal{L}$,$k$,$\nu$);
        $\boldsymbol{W}_i \leftarrow$ Train($\mathcal{L} \cup S_i, \mathcal{A}_i$)
    end for;
    return $(\boldsymbol{W}_1, \mathcal{A}_1), \ldots, (\boldsymbol{W}_\alpha, \mathcal{A}_\alpha)$;
```

Fig. 2. Algorithm for creating the ensemble

and biases of the network. Thus $\mathcal{A}$ and $\boldsymbol{W}$ together will specify a trained network. The output of Create_Ensemble is α trained networks. The decision on a test point is taken by a majority vote of these α networks.

The algorithm Train takes in two user defined parameters, k and ν. k is the parameter for the k nearest neighbor density estimation procedure. Choosing a proper value of k is a classical problem which do not yet have an well accepted solution, but there exist solutions (some very complicated) which solves this problem [7]. In the current work we do not attempt to solve this problem. In the next section we present some simulation results using this algorithm, we tested with numerous small values of k, we found that the performance do change with the change of k, but we did not find any significant pattern which shows a conclusive dependence of the parameter k with the performance. Based on experiments we suggest a value of k near 5. The parameter ν decides the number of new points that are to be included in each training set which is used for training the candidates. A small value of ν will mean little variation among the training set, and a big value of ν will mean more variability. But, the new points generated by Expand are noisy versions of the original training set, so a very big value of ν is not recommended. Our experiments suggest that ν being 10% of the size of the original training data gives good results.

The computational overhead in creating the ensemble is same as bagging except that it has the additional overhead of the function Expand. Expand requires finding the k nearest neighbors of each data point for computing the value d_i, this operation is computationally costlier than other operations involved. But the computation of the values d_i are a one time operation and they are not required to be repeated when Expand is called on the same training data multiple times. Thus, the total computational cost in creating the ensemble is not significantly more than that of conventional bagging.

5 Experimental Results

We tried our method on six real data sets from the UCI repository [1]. The data sets used are Iris, Wine, Liver-Disorder(Liver), Waveform-21(Wave), Pima-Indian-Diabetes (Pima), and Wisconsin Breast cancer (WBC). For the experiments we used the multilayered perceptron implementation of MATLAB. In particular we used the Levenberg-Marquardt backpropagation algorithm implemented as 'trainlm' method in MATLAB for training.

Table 1. The results

Data set	Single network	Conventional Bagging	Proposed Method			
			$k = 3$	$k = 5$	$k = 7$	$k = 9$
Iris	91.26±6.11	96.08±2.66	96.46±0.54	97.00±0.47	96.67±0.44	96.86±0.32
Wine	92.02±4.86	97.18±1.88	**98.93±0.32**	**98.70±0.38**	**98.70±0.38**	**99.04±0.53**
Liver	64.85±3.21	67.60±1.74	68.63±1.26	68.95±2.24	68.78±1.28	**68.95±1.70**
Wave	62.74±5.93	84.10±1.88	**86.09±0.18**	**86.43±0.32**	**85.98±0.26**	**85.70±0.24**
Pima	66.35±5.14	75.11±1.06	**77.03±0.83**	**76.66 ± 0.53**	**76.97±0.52**	**76.94±0.48**
WBC	95.71±0.54	96.37±0.44	95.98±0.41	96.06±0.34	96.10±0.33	95.86±0.36
Glass	62.06±3.53	67.66±1.78	**70.70±1.67**	**70.42±1.66**	**69.75±1.82**	**70.18±1.55**

Each of the results reported are for an MLP with 10 nodes in a single hidden layer. Each node has a sigmoidal activation function. Though we agree that this is not supposed to be 'optimal' for all cases. We could have used a validation set for determining the proper number of hidden unit for each data set. But, here our objective is to show that our method performs better than conventional bagging. So we decided to keep the number of hidden units and the number of hidden layer to be fixed across runs irrespective of the data sets. Same decision was taken with respect to the number of candidates in the ensemble. We fixed the number of members in the ensemble to be 10 for all cases. For all the data sets we take ν equal to 10% of the size of the training data.

The performance results reported are for a 10 fold cross validation repeated 10 times. The figures in Table 1 give the average performance and the standard deviation (in percentage) for six different scenarios. The performance of a single network, that of conventional bagging and that of our proposed method using $k = 3, 5, 7, 9$.

Table 1 clearly shows that the proposed method gives better results than conventional bagging for almost all data sets. The amount of improvement for some data sets are statistically significant. The figures are shown in bold if the performance of the proposed method is significantly better than conventional bagging[1].

6 Conclusion

We demonstrated a new method of creating ensembles. Our experiments demonstrates that the method shows improvements over conventional bagging for most of the data sets tried. We plan to address the following problems in future:

1. The procedures Expand and Train are quite general and can be used to train other kinds of classifiers other than a MLP. We plan to apply the method for other classifiers, in particular decision trees seem to be a good alternative.
2. Quantify the diversity among the candidates that this method yeilds.

[1] These results are based on a studentized t-test with 95% confidence.

Acknowledgements. This work was partially supported by CONACYT project I0013/APOY-COMPL-2008/90775.

References

1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Chen, R., Yu, J.: An improved bagging neural network ensemble algorithm and its application. In: Third International Conference on Natural Computation, vol. 5, pp. 730–734 (2007)
4. Drucker, H., Schapire, R.E., Simard, P.: Improving performance in neural networks using a boosting algorithm. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) NIPS, pp. 42–49. Morgan Kaufmann, San Francisco (1992)
5. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. CRC Press, Boca Raton (1993)
6. Georgiou, V.L., Alevizos, P.D., Vrahatis, M.N.: Novel approaches to probabilistic neural networks through bagging and evolutionary estimating of prior probabilities. Neural Processing Letters 27(2), 153–162 (2008)
7. Ghosh, A.K.: On optimum choice of k in nearest neighbor classification. Computational Statistics & Data Analysis 50(11), 3113–3123 (2006)
8. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. 12(10), 993–1001 (1990)
9. Kuncheva, L.I.: Diversity in multiple classifier systems. Information Fusion 6(1), 3–4 (2005)
10. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51(2), 181–207 (2003)
11. Mitra, P., Murthy, C.A., Pal, S.K.: Density-based multiscale data condensation. IEEE Trans. Pattern Anal. Mach. Intell. 24(6), 734–747 (2002)
12. Schapire, R.E.: A brief introduction to boosting. In: Dean, T. (ed.) IJCAI, pp. 1401–1406. Morgan Kaufmann, San Francisco (1999)
13. Schapire, R.E.: Theoretical views of boosting. In: Fischer, P., Simon, H.U. (eds.) EuroCOLT 1999. LNCS (LNAI), vol. 1572, pp. 1–10. Springer, Heidelberg (1999)
14. Zhou, Z.-H., Wu, J., Tang, W.: Ensembling neural networks: Many could be better than all. Artificial Intelligence 137(1-2), 239–263 (2002)

Leaks Detection in a Pipeline Using Artificial Neural Networks

Ignacio Barradas*, Luis E. Garza, Ruben Morales-Menendez,
and Adriana Vargas-Martínez

Tecnológico de Monterrey, Campus Monterrey
Ave. Garza Sada 2501, 64,849, Monterrey, NL, México
{A00780799,legarza,rmm,A00777924}@itesm.mx

Abstract. A system based on Artificial Neural Networks (*ANN*) is proposed to detect and diagnose multiple leaks in a pipeline leaks by recognizing the pattern of the flow using only two measurements. A nonlinear mathematical model of the pipeline is exploited for training, testing and validating the *ANN*-based system. This system was trained with tapped delays in order to include the system dynamics. Early results demonstrate the effectiveness of the approach in the detection and diagnosis of simultaneous multiple faults.

Keywords: Leak detection, Fault detection, Diagnosis, Artificial Neural Network.

1 Introduction

Distribution of fluids in pipelines must occur under safe and trustable conditions, because damages may be caused by environmental and weather conditions, as well as aging or pressure changes. Pipelines are design to support impacts or internal over pressure, but occasionally pressure surges may lead to line breaks and leaks. In some cases, pipelines are underground or in the sea depths. And to complicate the scenario even more, normally the fluids in transport do not operate under steady state conditions, which makes more difficult to perform a fault inspection. Additionally, small leaks are harder to detect, because they are a consequence of corrosion and aging in the pipeline.

There are three approaches for leak isolation have been proposed (internal, external and hybrid). The first approach (internal) is based on physical models, such as, mass and volume balance, pressure analysis and real-time dynamic models. The second approach (external) is implemented using hardware, such as, sensors with impedance or capacitance changes, optic fiber, gas sniffing, acoustic sensors, ground analysis or infrared image. And the third approach is hybrid methods, which are a mixture between internal and external approaches, for example: acoustic and pressure analysis with mass and volume balance.

[*] Any mail concerning this paper, should be send to the author's mail.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 637–644, 2009.

Leak isolation is still affected by expensive, noisy and vague instrumentation, uncertainties of the analytical model, and the relation between the operating point and leaks magnitude. A practical requirement for an automatic supervision system must be to detect the precise leak location as soon as possible and with a minimal amount of instrumentation.

In [1] several technologies to solve leak location applying automatic leak finders in pipelines where flow and pressure head instrumentation can be implemented only in their extremes is developed. In [3], a bank of observers with fixed leak positions satisfy the leak isolation and detection only if the pipeline is divided in three sections and two leaks are induced. The approach presented in [4] is the design of a parametric model in steady state which reduces the search interval. Although, these methods assure leak detection and diagnosis, they require intense mathematical formulation and a wide knowledge of the process. By this, there is a special interest in the application of Artificial Neural Networks (ANN) for solving fault diagnosis problems because of their classification and function approximation capabilities. ANN approach is convenient when an analytical model is difficult to obtain. In addition, ANN are highly robust to noisy inputs and to missing or new input data. Additionally, because of its parallel structure, ANN-based systems can be implemented for real time applications.

Recently, ANN-based approaches have taken special attention. In [2], two ANN cascade architecture were proposed, demonstrating that it is possible to detect leaks in pipeline. This work did not consider transient response when leaks occur. [6] demonstrated that using a neural-fuzzy system in a water distribution system makes possible to detect and classify faults in pipelines. A drawback of this approach is that multiple meters and gauges are needed in order to obtain the required information. Also, [5] used a fuzzy classifier to detect leaks in pipelines; this method used the transient response of the fluid in the pipeline, which requires very precise and continuous measurements. Finally, [10] compared several approaches for this application versus ANN-based systems.

This paper presents a method for detecting and isolating leaks in a pipeline. This method uses an ANN-based approach that recognizes the flow pattern using only two measurements. A mathematical model was proposed based on experimental data[1].

The paper is organized as follows. In Section 2 the pipeline model is described. Section 3 presents the proposed scheme. In section 4 the testing procedure that validates the approach and results are shown. And finally, Section 5 concludes the paper.

2 Pipeline Model with Leaks

In [7],[8], [9] and [11] the following mathematical model was introduced. The model was also validated with experimental data. The dynamic of the fluid through the pipeline is given by:

[1] Thanks C. Verde because her support with experimental data.

$$\frac{\partial Q}{\partial t} + gA\frac{\partial H}{\partial z} + \mu|Q|Q = 0 \qquad b^2\frac{\partial Q}{\partial z} + gA\frac{\partial H}{\partial t} = 0 \tag{1}$$

where H is the pressure head (m), Q is the flow (m^3/s), z is the length coordinate (m), t is the time coordinate (s), g is the acceleration of the gravity (m^2/s), A is the cross-section area (m^2), D is the pipeline diameter (m), b is the speed of sound (m/s), and $\mu = f/2DA$ where f is the *Darcy-Weissbach* friction coefficient.

A leak in point z_l will cause a discontinuity in equations (1) $Q|_{z_l} = \lambda_i\sqrt{H|_{z_l}}$ where $\lambda_i > 0$ is a function of the orifice area and discharge coefficient [12]. Because of this, a pipeline with $n-1$ leaks will be described by n pairs of differential equations, similar to equations (1) with a frontier condition between each pipeline segment given by:

$$Q^b|_{z_l} = Q^a|_{z_l} + Q|_{z_l} \tag{2}$$

where $Q^b|_{z_l}$ and $Q^a|_{z_l}$ are the flows before and after the leak. Having a pipeline of length L and assuming that the leaks are equally distributed along the space z, which can be divided in n segments of length $\Delta_z = L/n$. It is possible to approximate the partial derivatives of the pressure and flow with respect to the spatial variable z as follows:

$$\frac{\partial H}{\partial z} \cong \frac{H_{i+1} - H_i}{\Delta z} \qquad \frac{\partial Q}{\partial z} \cong \frac{Q_i - Q_{i-1}}{\Delta z} \tag{3}$$

where, the index i is associated with the variables at the beginning of the section i, and the frontier condition for each section is described by:

$$Q_{i+1} = \lambda_i\sqrt{H_{i+1}} \tag{4}$$

Knowing that the frontier conditions are characterized by the pressure H_{ri} and H_{ro}, at the beginning and the end of the pipeline, and substituting equations (3) in equations (1), the model could be described as a set of n coupled nonlinear equations given by:

$$\frac{\partial Q_i}{\partial t} = a_1(H_i - H_{i+1}) - \mu|Q_i|Q_i \qquad \frac{\partial H_i}{\partial t} = a_2(Q_{i-1} - Q_i) - (\lambda_{i-1}\sqrt{H_i})u_{t_i} \tag{5}$$

with $H_1 = H_{ri}$ and $H_{n+1} = H_{ro}$ as system inputs, and parametric constants $a_1 = g\pi r^2 n/L$ and $a_2 = b^2L/g\pi r^2 n$ with $n = 4$, $u_{t_i} = u(t - t_i)$ is the unit step function associated with the occurrence time t_i of the leak i.

If leaks are not equally distributed, Δz is not constant and parameters a_1 and a_2 are function of the distance between leaks. For this study case, H_1 and H_5 are constant pressures heads at the input and output of the pipeline, while flows Q_1, and Q_4 are the measurable flows at the extremes of the pipeline. For this study, only three leaks were considered each one at the frontier condition, Fig. 1.

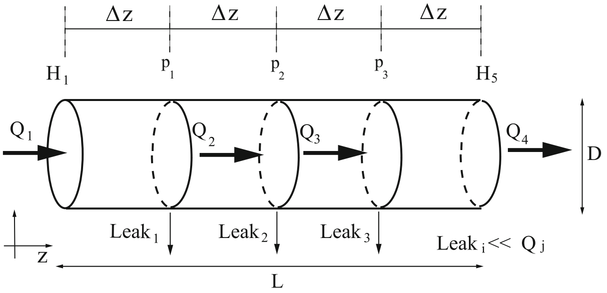


Fig. 1. Pipeline discrete model [10]

3 Proposed Scheme

Fig. 2 shows the implemented scheme. The scheme consists of an *ANN* that detects the leak and its location in the pipeline. Tapped delays signal from Q_1 (flow measurement at the inlet) and Q_4 (flow measurement at the outlet) were introduced as inputs of the *ANN* in order to include the system dynamics. The above resulted as an improvement of the *ANN* performance.

Many *ANN* configurations were tested. All of them were feed-forward multilayer architecture. The classical back-propagation algorithm was used for the learning step. The basic differences in each *ANN* configuration were the number of neurons and layers; however, the number of delays in the input signals took the highest impact in the results. The input layer neurons use a tan-sigmoid activation function and the output layer neurons use a log-sigmoid function.

The leak detector is mainly based on the *ANN* performance. The *ANN* uses only the inlet/outlet flow measurements. The detector system identifies the possible pipeline operating states. And based on the state, the leak can be detected.

The *ANN* output will generate a leak signature according to Table 1, and it will be translated into an operating state by the state codifier. This codifier is based on simple logic rules, which will assign a state due to the outputs generated by the *ANN*.

The number of operating states that the codifier can estimate is given by the number of sections in which the pipeline is segmented. For this case, the pipeline was split in three segments; therefore, there are eight operating states, Table 1.

Table 1. Operating states of the pipeline

State	Activated Leaks	f_1	f_2	f_3	State	Activated Leaks	f_1	f_2	f_3
1	No leaks	0	0	0	5	1 and 2	1	1	0
2	1	1	0	0	6	1 and 3	1	0	1
3	2	0	1	0	7	2 and 3	0	1	1
4	3	0	0	1	8	1, 2 and 3	1	1	1

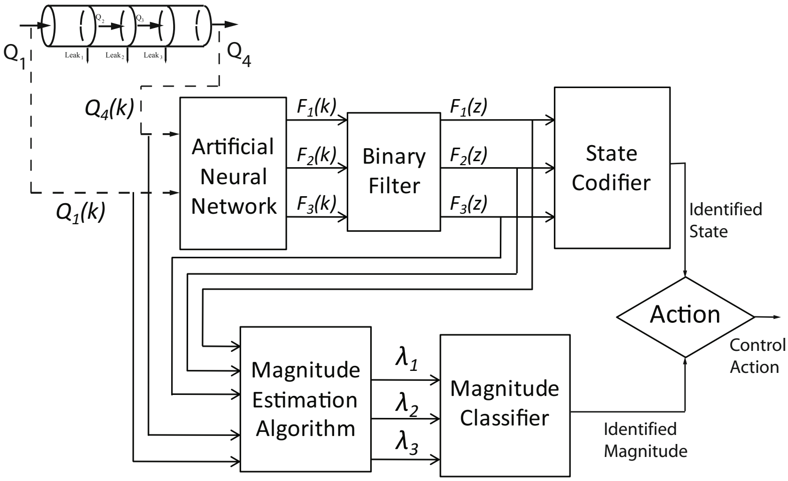


Fig. 2. Detection and diagnosis scheme

It is important to notice that the outputs of the *ANN* are activated by a hyperbolic tangent function; this means that the output can take a value between 0 and 1. A binary fault signature is needed, therefore a filter is included.

After estimating the operating state, the segment or segments of the pipeline in which the fault occurs are found. The accuracy of this approach will depend on the number of segments used in the pipeline.

The *ANN* was trained with information about all possible leaks. All the possible state transitions are included in the training step based on a Markov chain simulation. The input data for the training step were the inlet/outlet flows and their delays; the output data was the leak signature. It is important to mention, that a better *ANN* training is possible if variations in the discharge coefficients are introduced in the generation of the training data set.

4 Results

The proposal approach was validated with 4 tests. *Test-1* was the introduction of never seen-before data to the *ANN*. *Test-2* consisted in adding noise (N) to the flow signals. *Test-3* corresponds of two experiments for testing the robustness. First, the nominal pressure was changed from $H_1 = 11$ m and $H_5 = 5$ m to $H_1 = 14$ m and $H_5 = 8$ m; second, the input data was generated with variations in the value of the discharge coefficients (λ). Finally, in *Test-4* the *ANN* was re-trained with a training data set that includes variations in the discharge coefficient.

The performance index corresponds to the error generated between the real states of the pipeline (multiple possible scenarios generated by the simulator) and the estimated states computed by the *ANN*.

Fig. 3 displays three plots, all of them represent the activation of leak 1. Top plot correspond to the real operating states, middle plot shows the estimated

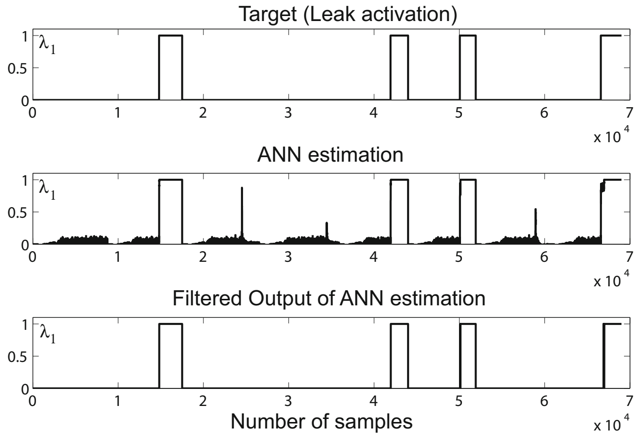


Fig. 3. Outputs from the ANN. Real state of pipeline; ANN output; filtered ANN output.

states, and the bottom plot represent the filtered states. This last signal plot is the one entering to the state codifier; and in combination with the other two leak signals the operating state of the pipeline is determined.

Fig. 4 shows the operating states estimations. This results proof that the proposal approach gives acceptable predictions of the real conditions in the pipeline.

Table 2 summaries the results. This table shows the error in the training process, and then the *ANN* configuration in each experiment depends on the flow signal delays. As it can be seen, in *Test-2* two different noise levels are added, 0.01% and 0.015%. As shown in the Table, the *ANN* that considers more delays

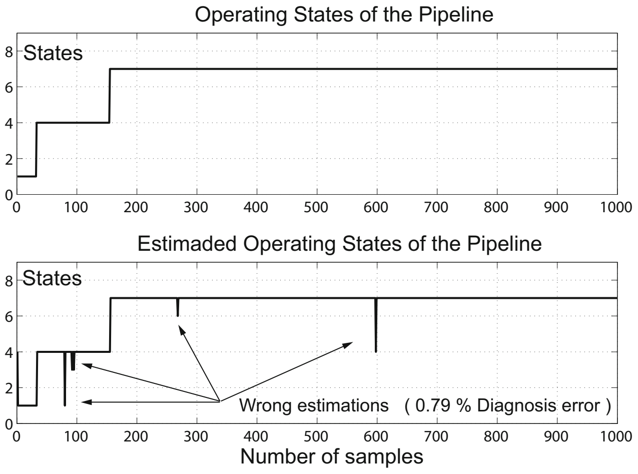


Fig. 4. State Estimation by the System. Real state of pipeline; ANN prediction.

Table 2. Summary of the results obtained after the experimentation

ANN	Error	Network Architecture	Test 1	Test 2		Test 3		Test 4
				$N = 0.01\%$	$N = 0.015\%$	Set point	$\lambda(\%)$	
ANN_1	0.00218	2-10-10-3	1.37%	1.37%	-	3.39%	27	4.53%
ANN_2	4.9×10^{-8}	4-4-4-3	0.4%	3.8%	6.39%	3.79%	3.09	-
ANN_3	4.4×10^{-8}	6-6-6-3	0.0001%	0.8%	2.4%	1.4%	12.7	6.13%

a better performance. As expected, having more information about the dynamic behaviour helps the ANN to recognize the pattern, even with a noisy signal. In *Test-3*, two validation experiments were conducted; first the head pressures were modified (set point), and second the size of the leaks were changed. These modifications impact directly the transient response of the flow (a new pattern is generated), as it can be seen in Table 2; even though, the performance of the ANN was acceptable.

The diagnosis error index (wrong estimated states/ total real states) was used as a main indicator of the performance of each ANN. It is important to notice that ANN_2 and ANN_3 used as inputs tapped delays. ANN_2 used one delay, while ANN_3 used 2 delays for each input signal.

[2] proposed a similar scheme for leak detection. In that research multiple sensors information was used in order to detect one and two leaks. In the present work similar results were obtained using only 2 measurements: inlet and outlet flow signals. The utilization of delayed signal allowed the algorithm to learn from the transient response, and therefore it was not necessary to use information of intermediate points in the pipeline.

It is important to mention that using this model allows the simulator to divide the pipeline in many segments. If there exist more segments, more data and operating states will be created as a consequence and therefore the accuracy of the leak location will be improved.

5 Conclusions

The main contribution of this work is that proves to be possible to estimate the location of the leak or leaks by only measuring the inlet and outlet flow, disregarding the pressure measurements and the size of the leak needed in [2], this can be observed in Fig. 3, and in Table 2, in which the ANN errors are $0.00218, 4.9\times10^{-8}$ and 4.4×10^{-8} for the ANN_1, ANN_2 and ANN_3, respectively.

Using tapped delays as inputs of the network demonstrated to improve the ANN estimations, because the network is capable of learning the flow's dynamic behavior. And this dynamic makes possible to identify the pattern and the different transitions between the operational states. In addition, it can be seen how the ANN using tapped delays has a smaller detection error, and also are more efficient in computer cost; require less training time and achieve a better training error value. Also, it is important to notice that the ANN_2 and ANN_3 were more sensitive to the discharge coefficients variation, due to the fact that they learn the transient response of the fluid dynamic (Table 2).

The training data set in which the network is trained is decisive. In order to generate a robust network, choosing a set that includes as many operational scenarios as possible is necessary. Therefore an acceptable detection scheme would be created. This can be observed in test 4, in which a better design set of training data were introduced to the networks.

References

1. Ashton, S.A., Shields, D.N., Daley, S.: Fault Detection in Pipelines using Nonlinear Observers. In: UKACC Int. Conf. on Control IEE Conf., vol. 455, pp. 135–140 (1998)
2. Belsito, S., Lombardi, P., Andreussi, P., Banerjee, S.: Leak Detection in Liquefied Gas Pipelines by Artificial Neural Networks. AIChE J. 44(12), 2675–2688 (1998)
3. Billman, L., Issermann, R.: Leak Detection Methods for Pipelines. Automática 23(3), 381–385 (1987)
4. Blanke, M., Frei, Ch., Kraus, F., Patton, R.J., Staroswiecki, M.: What is Fault Tolerant Control? Safeprocess 35, 123–126 (2000)
5. Crowther, W.J., Edge, K.A., Burrows, C.R., Atkinson, R.M., Wollons, D.J.: Fault Diagnosis of a Hydraulic Actuator Circuit using Neural Networks an Output Vector Space Classification approach. Proc. Inst. Mech. Eng. Part I: J. Syst. Control Eng. 212(11), 57–68 (1998)
6. Izquierdo, J., López, P.A., Martínez, F.J., Pérez, R.: Fault Detection in Water Supply Systems using Hybrid Modelling. Mathematical and Computer Modelling 46, 341–350 (2007)
7. Verde, C.: Multi-leak Detection and Isolation in Fluid Pipelines. Control Eng. Practice 9, 673–682 (2001)
8. Verde, C.: Accommodation of Multi-leaks Positions in a Pipeline. Control Eng. Practice 13, 1071–1078 (2005)
9. Verde, C., Visairo, N., Gentil, S.: Two Leaks Isolation in a Pipeline by Transient Response. Advances in Water Resources 30, 1711–1721 (2007)
10. Verde, C., Morales-Menendez, R., Garza, L.E., De La Fuente, O., Vargas-Martínez, A., Velasquez, P., Aparicio, C., Rea, C.: Multi-Leak Diagnosis in Pipelines - A Comparison of Approaches. In: Special Session of the Mexican Int. Conf. on Artificial Intelligence, pp. 352–357 (2008)
11. Visairo: Detección y Localización de Fugas en un Ducto, PhD Thesis, SEP-CENIDET, México (2004)
12. Zhidkova, M.A.: Gas Transportation in Pipelines. In: Internal report written in Russian. Naukov, Dumka, USSR (1973)

Computing the Weights of Polynomial Cellular Neural Networks Using Quadratic Programming

Anna Rubi-Velez[1], Eduardo Gomez-Ramirez[2], and Giovanni E. Pazienza[3]

[1] Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Quatre Camins 2,
08022 Barcelona (Spain)
[2] LIDETEA, Posgrado e Investigacion, Universidad La Salle, Benjamin Franklin 47,
Col. Condesa, 06140 Mexico City (Mexico)
[3] Cellular Sensory and Wave Computing Laboratory
MTA-SZTAKI, Budapest (Hungary)
st12961@salle.url.edu,
egr2@ulsa.mx,
gpazienza@sztaki.hu

Abstract. Finding the weights of a Polynomial Cellular Neural/Nonlinear Network performing a given task is not straightforward. Several approaches have been proposed so far, but they are often computationally expensive. Here, we prove that quadratic programming can solve this problem efficiently and effectively in the particular case of a totalistic network. Besides the theoretical treatment, we present several examples in which our method is employed successfully for any complexity index.

Keywords: polynomial cellular neural networks, cellular automata, quadratic programming.

1 Introduction

Since the origin of the Cellular Neural Networks [1] in 1988, there is a natural concept relation with the Cellular Automata (CA) [2]. However, in spite of this relation, there are few papers that describe in a formal way the mathematical relations between them. Recently, it has been shown how nonlinear dynamics can give a new perspective of CA, by proving that Cellular Automata are a particular case of a more general paradigm called Cellular Neural Networks (CNNs) [3]. This approach has shed new light on CA, for which novel concepts – e.g., the index of complexity for one-dimensional CA – have been introduced. As proved in [4], the bridge between Cellular Automata and Cellular Neural Networks is the so-called Universal CNN cell (also known as Generalized CA). Nevertheless, the dynamic behavior of a Universal CNN cell can be also synthesized by means of other nonlinear functions, such as polynomials. On this ground, we put forward Polynomial CNNs (PCNNs), a cellular paradigm that has been already applied to several problems, including the exclusive OR and the Game of Life [5]. The main advantage of this model is that even the simplest implementation, single-layer and space-invariant weights, has the same computational power as a Universal Turing Machine but with the computational cost of the learning using genetic algorithm. In [6] the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 645–652, 2009.

authors present the relationship between polynomial CNNs and two-dimensional CA providing a formal method to find the weights of the network through a simple procedure based on the properties of polynomials. The problem in this methodology is that is not easy to find the template if the order of the polynomial is more than 3. In this paper a new simple set of equations are presented to find the weights for any order of the polynomial using quadratic programming. With this model it is possible to find minimal solutions using the quadratic metric of the weights. The advantage is that for the specific case of modeling totalistic cellular automata it is not necessary to use genetic algorithm or other computational technique. The structure of the paper is the following: section 2, describes some cellular automata concepts; in section 3, PCNN's are introduced and in section 4, the quadratic programming approach is presented for any order of the PCNN. Finally section 5presents some conclusions.

2 Cellular Automata

Cellular Automata consist of regular uniform lattice of cells assuming a finite number of states; here, we consider two-dimensional CA in which cells are arranged in an eight neighbor rectangular grid and can take only two states. Cells are updated synchronously (discrete-time evolution) and the state of each cell at iteration $n + 1$ depends on the states of the cells in its neighborhood at iteration n. Similarly as before, we consider that the neighborhood of a cell is composed by the cell itself plus its eight nearest neighbors. Therefore, a two-dimensional Cellular Automaton is a discrete-time system with 9 binary inputs and 1 binary output; consequently, the dynamics of CA can be conveniently represented on a truth table containing $2^9 = 512$ rows. In total, there are $2^{2^9} = 2^{512}$ of such tables, also called rules, corresponding to all possible ways to evolve two-dimensional binary CA. We assume without loss of generality that at any fixed time the value of each cell can be either +1 (active cells) or -1 (non-active cells), exactly as in CNNs.

2.1 Totalistic Cellular Automata

The definition of a Totalistic Cellular Automaton is that the next state of a cell depends exclusively on the number of active cells in the neighborhood at the previous state. This means that, the position of the active cells in the neighborhood does not influence the result. The truth tables defining the rules of totalistic CA (see Table 1) have only 10 rows corresponding to all possible combinations of active and non-active cells in the neighborhood: from 0 (no active cell in the neighborhood) to 9 (all cells in the neighborhood are active).

There are only $2^{10} = 1024$ totalistic CA rules, (from $N = 0$ to $N = 1023$). The rule number is given by the following formula:

$$N = \sum_{i=0}^{9} \left(\frac{\beta_i+1}{2}\right) 2^i \qquad (1)$$

where N is the rule number, and the values of β can be retrieved from table 1. Rules of totalistic CA can be conveniently represented in a Cartesian coordinate system. This

Table 1. Truth table for a Totalistic Cellular Automata Rule

Active neighbors (Σ neighbors)	Output	Rule 51
0 (-9)	β_0	1
1 (-7)	β_1	1
2 (-5)	β_2	-1
3 (-3)	β_3	-1
4 (-1)	β_4	1
5 (+1)	β_5	1
6 (+3)	β_6	-1
7 (+5)	β_7	-1
8 (+7)	β_8	-1
9 (+9)	β_9	-1

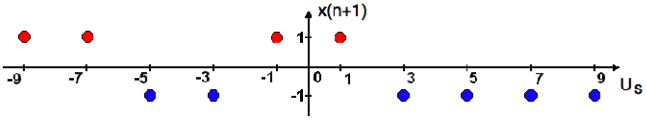


Fig. 1. CA dynamic diagram for totalistic rule 51

original representation was proposed in [6], in which the sum of the nine neighbors of the automaton is on the horizontal axis, and the corresponding output on the vertical axis. Given a rule (or equivalently, a truth table), for each of the 10 input patterns we need to depict a red, in case of a firing pattern; blue, in case of a quenching pattern. For example, the rule corresponding to the truth table in Table 2 is:

$$N = \sum_{i=0}^{9} \left(\frac{\beta_i + 1}{2}\right) 2^i = 1 + 2 + 16 + 32 = 51 \tag{2}$$

and its dynamic diagram is in Fig. 1

3 Polynomial Cellular Neural Network

The Polynomial Cellular Neural Networks (PCNN) was first introduced in [7] and other models can be found in [8]. The general form for PCNN in discrete time is:

$$x(n+1) = A * Y(n) + B * U + z + P_2(u, y) + \cdots + P_m(u, y) \tag{3}$$

where *m* is the order of the PCNN. Considering the previous work [6], it is possible to find through a rigorous method the parameters – degree of the polynomial and the weights of the network – of a one-layer space-invariant Polynomial Cellular Neural Network implementing a totalistic CA rule. When implementing a totalistic CA, the mathematical representation can be simplified thanks to two preliminary considerations about the nature of the problem. First of all, since the output of a CA local rule depends exclusively on the input, also the output of the network has to be a function of the input pattern only. This means that no matrix, except for those convolving the input U and its multiples, can have non-zero elements other than the central one. In other words, in the model of previous equation all matrices convolving the output *Y* and its multiples must have the central element only. For instance,

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} = a \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

and, consequently:

$$A * Y = ay_c \tag{5}$$

where the subscript *c* emphasizes that the only value to take into account is the central one. In totalistic CA all neighbors are considered at once, and hence there is also no reason for making a distinction among the different values of the matrices convolving the input U and its multiplies. For instance,

$$B = \begin{bmatrix} b & b & b \\ b & b & b \\ b & b & b \end{bmatrix} = b \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{6}$$

and, hence the convolution $B * U$ can be simplified as:

$$B * U = b \sum_{(k,l) \in N(i,j)} u_{kl} = bu_s \tag{7}$$

where the subscript *s* emphasizes that the value u_s results from the sum of all nine neighbors. In conclusion, Eq. 3 can be rewritten as a function of u_s and y_c as follows:

$$x(n+1) = ay_c(n) + bu_s + z + P_2(u_s, y_c) + \cdots + P_m(u_s, y_c) \tag{8}$$

where $u_s \in \{-9,-7,-5,-3,-1, 1, 3, 5, 7, 9\}$, which are the values obtainable by summing nine values ±1, and $y_c \in \{-1,+1\}$, by definition of discrete-time PCNN.

3.1 First Order PCNN

The first order PCNN is equivalent to a standard CNN, whose state equation, using the notation just introduced, is:

$$x(n+1) = a_1 y_c(n) + b_1 u_s + z \tag{9}$$

where the subscript 1 added to the network weights *a* and *b* means that we consider a first order model.

$$y_\infty = \begin{cases} 1 \Leftrightarrow b_1 u_s + z \geq 0 \\ -1 \Leftrightarrow b_1 u_s + z < 0 \end{cases} \tag{10}$$

From the previous equations, it is easy to see that when $a_1 > 0$ the network is stable, and it converges to the steady state y_∞.

Choosing properly the weights b_1 and *z*, Eq. 9 describes any line. Therefore, a first degree PCNN can solve only linearly separable problems, or equivalently, perform all linearly separable totalistic CA rules. Note that given a totalistic CA rule *N* there are infinite ways of setting b_1 and *z* to obtain it; however, fixed values of b_1 and *z* define univocally a totalistic CA rule. An example of application of first degree PCNN is illustrated in Fig. 2.

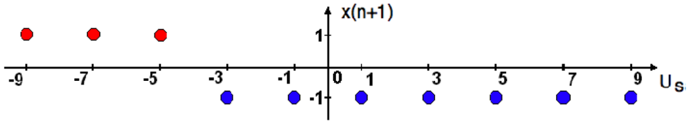


Fig. 2. Rule 7 can be implemented by a first order Polynomial CNNs

3.2 Second Order PCNNs

The generic form for the second degree polynomial $P_2(u_s, y_c)$ is:

$$\begin{aligned}\mathrm{P}_2(u_s, y_c) &= \sum_{i=0}^{2} p_i u_s^i \cdot q_i y_c^{2-i} \\ &= p_0 u_s^0 \cdot q_0 y_c^2 + p_1 u_s \cdot q_1 y_c + p_2 u_s^2 \cdot q_2 y_c^0 \\ &= p_1 q_0 + p_1 q_1 u_s y_c + p_2 q_2 u_s^2\end{aligned} \tag{11}$$

where in the last step we used the property that $u_s^0 = 1$ and $y_c^i = 1$, for i even, and $y_c^i = y_c$, for i odd. The term $p_0 q_0$ is constant and hence it can be absorbed into the bias z (with abuse of notation, $z + p_0 q_0 = z$) and the expression can be further simplified through the substitutions $p_1 q_1 = a_2$ and $p_2 q_2 = b_2$. Finally, we can add this expression to previous equation obtaining:

$$\begin{aligned}& x(n+1) = a_1 y_c(n) + b_1 u_s + z + a_2 u_s y_c(n) + b_2 u_s^2 \\ & = (a_2 u_s + a_1) y_c(n) + b_2 u_s^2 + b_1 u_s + z\end{aligned} \tag{12}$$

where $a_1, a_2, b_1, b_2, z \in \Re$. If the term $(a_2 u_s + a_1) > 0$, the network is always stable, and it converges to the steady state y_∞

$$y_\infty = \begin{cases} 1 & \Leftrightarrow b_2 u_s^2 + b_1 u_s + z \geq 0 \\ -1 & \Leftrightarrow b_2 u_s^2 + b_1 u_s + z < 0 \end{cases} \tag{13}$$

With this model, the PCNN is capable of implementing any totalistic CA rule whose firing and quenching patterns in the CA dynamic diagram can be separated by two lines. An example of application of second order PCNN is in Fig. 3: this CA dynamical diagram describes the behavior of rule 12.

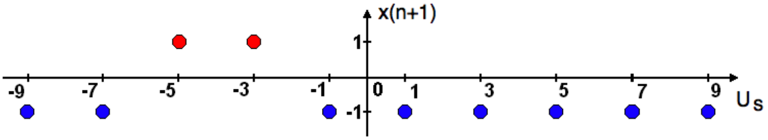


Fig. 3. Rule 12 can be implemented by a second order Polynomial CNNs

3.3 Third Order PCNNs

With the same considerations as before about u_0 and y_i, and through the substitutions, the general equation for the third order PCNN implementing totalistic CA can be described as:

$$\begin{aligned}& x(n+1) = (a_2 u_s + a_1) y_c(n) + b_2 u_s^2 + b_1 u_s + z + \mathrm{P}_3(u_s, y_c) \\ & = (a_3 u_s^2 + a_2 u_s + a_1) y_c(n) + b_3 u_s^3 + b_2 u_s^2 + b_1 u_s + z\end{aligned} \tag{14}$$

Where $a_1, a_2, a_3, b_1, b_2, b_3, z_2 \in \Re$. Considering $a_3 u_s^2 + a_2 u_s + a_1 > 0$, which ensures the stability of the PCNN, the network output can be computed as:

$$y_\infty = \begin{cases} 1 & \Leftrightarrow b_3 u_s^3 + b_2 u_s^2 + b_1 u_s + z \geq 0 \\ -1 & \Leftrightarrow b_3 u_s^3 + b_2 u_s^2 + b_1 u_s + z < 0 \end{cases} \tag{15}$$

which means this kind of network is capable of implementing any totalistic CA rule whose firing and quenching patterns in the CA dynamic diagram can be separated by three lines. An example of application of second degree PCNN is illustrated in Fig. 1: according to the notation introduced before, this CA dynamical map describes the behavior of rule 51.

3.4 M-th Order PCNN and Continuous-Time Model

The procedure illustrated previously can be repeated for higher degree polynomials, obtaining similar results: the PCNN state equation of any order contains a term multiplying *$y_c(n)$,* controlling the dynamics of the network, and another term which depends only on us and the network weights, determining the steady state of the network. The last term can be used to interpolate the points of the CA dynamic pattern, and the resulting linear system has always a solution since it is Vandermonde-like. Therefore, we can conclude that the complexity index κ of a totalistic CA rule corresponds to the order of the PCNN model implementing it. It can be also noticed that the complexity index in totalistic CA is equal to the number of lines separating firing and quenching patterns in the CA dynamic diagram; hence, the minimum value for κ is 0 (rule 0 and rule 1023) and its maximum value is 9 (rule 341 and rule 682, see Fig. 6).

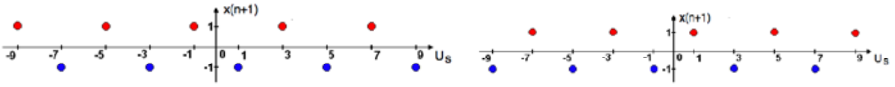


Fig. 4. Totalistic CA rules having complexity index κ = 9: (a) rule 341 and (b) rule 682

4 Computing the Weights Using Quadratic Programming

Defining the set of templates for a second order PCNN as:

$$w^T = \{a_2, a_1, b_2, b_1, z\}$$

Equation (13) can be translated to:

$$y_\infty = \begin{cases} 1 \Rightarrow b_2 \left(u_s^+\right)^2 + b_1 u_s^+ + z \geq 0 \\ -1 \Rightarrow b_2 \left(u_s^-\right)^2 + b_1 u_s^- + z < 0 \end{cases} \tag{16}$$

Where u_s^+ and u_s^- are the values that corresponds to outputs *x(n+1)* 1 and -1 respectively. For example, for rule 7, $u_s^+ = \{-9, -7, -5\}$ and $u_s^- = \{-3, -1, 1, 3, 5, 7, 9\}$.

For the general case it is possible to write the stability condition as:

$$\sum_{i=1}^{m} a_i u_s^{i-1} > 0 \tag{17}$$

and:

$$y_\infty = \begin{cases} 1 \Leftrightarrow \sum_{i=1}^{m} b_i u_s^i + z \geq 0 \rightarrow b_1 u_s^+ + b_2 (u_s^+)^2 + \ldots + b_m (u_s^+)^m + z \geq 0 \\ -1 \Leftrightarrow \sum_{i=1}^{m} b_i u_s^i + z < 0 \rightarrow -b_1 u_s^- - b_2 (u_s^-)^2 - \ldots - b_m (u_s^-)^m - z > 0 \end{cases} \tag{18}$$

Considering the previous equation, it is possible to define the following optimization quadratic programming problem:

$$\min_{w} w^T H w \qquad (19) \qquad\qquad \begin{bmatrix} A \\ A_e \end{bmatrix} w \begin{matrix} > \\ \geq \end{matrix} \begin{bmatrix} d \\ d_e \end{bmatrix} \qquad (20)$$

The first order case can be written in matrix form as:

$$\begin{bmatrix} u_s & 1 & 0 & 0 & 0 \\ 0 & 0 & -\left(u_s^-\right)^2 & -u_s^- & -1 \\ 0 & 0 & \left(u_s^+\right)^2 & u_s^+ & 1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ b_2 \\ b_1 \\ z \end{bmatrix} \begin{matrix} > \\ > \\ \geq \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

and the general case as:

$$\begin{bmatrix} u_s^{(m-1)} & u_s^{(m-2)} & \ldots & 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & -\left(u_s^-\right)^m & -\left(u_s^-\right)^{(m-1)} & \ldots & -u_s^- & -1 \\ 0 & 0 & \ldots & 0 & \left(u_s^+\right)^m & \left(u_s^+\right)^{(m-1)} & \ldots & u_s^+ & 1 \end{bmatrix} \begin{bmatrix} a_m \\ a_{m-1} \\ \ldots \\ a_1 \\ b_m \\ b_{m-1} \\ \ldots \\ b_1 \\ z \end{bmatrix} \begin{matrix} > \\ > \\ \geq \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{22}$$

With the previous representation is possible to use any algorithm to solve quadratic programming equations. Adapting the previous equation to the quadratic programming function of the Matlab, we use and ε approximation for the first two rows such as: $aw > 0 \rightarrow aw \geq \varepsilon, \forall \varepsilon \rightarrow 0$.

Table 2. Templates obtained using PCNN

Rule	m	ε	Weights
7	1	1e-9	1.0e-008* [0.1000 -0.0500 -0.2500]^T
12	2	1e-9	1.0e-008 *[0 0.1000 -0.0125 -0.1000 -0.1875]^T
51	3	1e-9	1.0e-009 *[0 0 1 -0.1042 -0.5629 0.1042 0.5629]^T
682	9	1e-9	1.0e-009*[0.3354 -0.1693 0.3354 -0.1693 0.3354 -0.1693 0.3354 -0.1693 0.3354 0 0 -0.0020 0 0.0593 0 -0.5079 0 0.9507 -0.5000]^T

5 Conclusions

The computational cost of polynomial cellular neural networks training can be improved using the previous methodology. For these cases is not necessary to use genetic algorithm to find the solutions. It is possible to use some variations of the linear or quadratic programming approach to find solutions in the integer domain or with a specific resolution. Using the results presented in [4] and with the results presented in this work, a similar approach can be obtained for semitotalistic cellular automata rules.

References

1. Chua, L.O., Yang, L.: Cellular neural networks: Theory and Applications. IEEE Trans. Circuits Syst. 35, 1257–1290 (1988)
2. von Neumann, J.: The general and logical theory of automata. In: Jeffress, L.A. (ed.) Cerebral Mechanisms in Behavior–The Hixon Symposium. s.l. John Wiley, Chichester (1951)
3. Chua, L.O., Yook, S., Dogaru, R.: A nonlinear dynamics perspective of Wolfram's new kind of science. Part I: Threshold of complexity. International Journal of Bifurcation and Chaos 12, 2655–2766 (2002)
4. Dogaru, R., Chua, L.O.: Universal CNN cells. International Journal of Bifurcation and Chaos 9(1), 1–48 (1999)
5. Pazienza, G.E., Gomez-Ramirez, E.y., Vilasis-Cardona, X.: Portugal Polynomial discrete time cellular neural networks for implementing the Game of Life.: s.n. In: Proc. International Conference on Artificial Neural Networks, ICANN 2007 (2007)
6. Giovanni, E., Gomez-Ramirez, E.: New properties of 2D Cellular Automata found through Polynomial Cellular Neural Network. In: International Joint Conference on Neural Networks (IJCNN 2009), Pazienza, Atlanta, USA (2009)
7. Schonmeyer, D., Feiden, D., Tetzlaff, R.: Multi-template training for image processing with cellular neural networks. In: Proc. of 2002 7th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA 2002), Frankfurt, Germany (2002)
8. Pazienza, G., Gomez-Ramirez, E., Vilasis-Cardona, E.: Polynomial discrete time cellular neural networks to solve the XOR problem. In: Proc. 10th International Workshop on Cellular Neural Networks and ther Applications (CNNA 2006), X. Istanbul, Turkey (2006)

Two-Dimensional Fast Orthogonal Neural Network for Image Recognition

Bartłomiej Stasiak

Institute of Computer Science, Technical University of Łódź
ul. Wólczańska 215, 93-005 Łódź, Poland
basta@ics.p.lodz.pl

Abstract. This paper presents a method of constructing a fast orthogonal neural network suitable for raw image recognition and classification. The neural architecture of the proposed network is based on fast cosine transform algorithm modified to enable Fourier amplitude spectrum computation. The presented network has reduced computational complexity and it reaches low generalization error values as compared to a standard multilayer perceptron.

Keywords: fast orthogonal neural network, Fourier amplitude spectrum, image recognition.

1 Introduction

Image recognition (IR) is one of the key fields of artificial intelligence applications and still an open one. On one hand, high dimensionality and redundancy of visual data necessitates search of effective feature extraction algorithms. On the other hand, the practically infinite variability of possible scene composition makes the choice of the features difficult in a general case. A satisfying level of invariance to distortions of a given type often requires that the features are carefully constructed by hand, which confines the autonomy of IR systems.

A radical solution is to skip the feature extraction stage, feeding the almost raw image data to the suitable classifier. This approach may yield good results if the size of the images is kept reasonably small and the training data is sufficiently rich [1]. However, increasing the input data dimensionality usually leads to overfitting the classifier, which results in growing generalization error.

The solution to this problem may be sought in changing the data model to e.g. tensor representation [2], or in controlling the classifier complexity which, in the case of a multilayer neural network, may mean reducing the size of the hidden layers. In this paper we propose a different approach: the architecture of a neural network based on a fast algorithm of two-dimensional Fourier transform. This general type of neural network is referred to as fast orthogonal neural network (FONN, [3]). Such a network consists of $O(logN)$ sparsely connected layers, each containing $O(N)$ neurons, where N is the input dimensionality. Hence, the neural weights reduction corresponds to the reduction of computational complexity typical of fast algorithms of orthogonal transforms.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 653–660, 2009.

Choosing Fourier transform as the basis for the FONN has an additional advantage: we can add a special output layer enabling to compute Fourier amplitude spectrum, which is often used in IR problems for feature extraction, due to some interesting properties (e.g. shift invariance). It should be noted, however, that what we propose is a *neural network* - it can learn how to compute Fourier transform in a fast way but it can also adapt for any other linear transform realizable with its sparse architecture. Therefore, we can think of an *adaptable* Fourier transform in which every basic operation of the fast algorithm is converted to a basic operation orthogonal neuron (BOON, [3]).

The connection scheme of the proposed network is based on fast homogeneous algorithm of two-dimensional cosine transform [4] which is modified to enable Fourier transform computation. The details of the necessary modifications are presented in the next section.

2 Neural Network Architecture

Our goal is to construct a fast computational scheme of Fourier transform which is defined for a discrete two-dimensional signal $x(n,m)$ as:

$$X_{N\times M}(p,q)=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}x(n,m)e^{-j2\pi\frac{pn}{N}}e^{-j2\pi\frac{qm}{M}} \ , \tag{1}$$

where j is the imaginary unit, $p,n=0,1,...,N-1$ denote the row number, $q,m=0,1,...,M-1$ denote the column number, and N, M define the height and width of the input image, respectively.

Every output value defined by eq. (1) is a complex number with real and imaginary part equal to:

$$\begin{aligned} Re\{X_{N\times M}(p,q)\} &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}x(n,m)\left(C_N^{pn}C_M^{qm}-S_N^{pn}S_M^{qm}\right) \ , \\ Im\{X_{N\times M}(p,q)\} &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}x(n,m)(-C_N^{pn}S_M^{qm}-S_N^{pn}C_M^{qm}) \ , \end{aligned} \tag{2}$$

where $C_K^k=\cos(2\pi k/K)$, $S_K^k=\sin(2\pi k/K)$. Our starting point to compute these values is two-dimensional discrete cosine transform, type II given as [5]:

$$L_{N\times M}^{II}(p,q)=\mathrm{DCT}_{N\times M}^{II}\{x(n,m)\}=\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}x(n,m)C_{4N}^{(2n+1)p}C_{4M}^{(2m+1)q} \ . \tag{3}$$

Utilizing formula (3) for computation of the values defined by eq. (2) requires a special permutation of input values $x(n,m)$:

$$\begin{aligned} x_T(2n,2m) &= x(n,m) \ , \\ x_T(2n+1,2m) &= x(N-1-n,m) \ , \\ x_T(2n,2m+1) &= x(n,M-1-m) \ , \\ x_T(2n+1,2m+1) &= x(N-1-n,M-1-m) \ . \end{aligned} \tag{4}$$

Let us consider the following decomposition of the cosine transform of the permuted signal $x_T(n,m)$:

$$\begin{aligned} L_T^{II}(p,q) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x_T(n,m)C_{4N}^{(2n+1)p}C_{4M}^{(2m+1)q} = \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x_T(2n,2m)C_{4N}^{(4n+1)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x_T(2n+1,2m)C_{4N}^{(4n+3)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x_T(2n,2m+1)C_{4N}^{(4n+1)p}C_{4M}^{(4m+3)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x_T(2n+1,2m+1)C_{4N}^{(4n+3)p}C_{4M}^{(4m+3)q} . \end{aligned} \tag{5}$$

Taking into account the formulae (4) we have:

$$\begin{aligned} L_T^{II}(p,q) = \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x(n,m)C_{4N}^{(4n+1)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x(N-1-n,m)C_{4N}^{(4n+3)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x(n,M-1-m)C_{4N}^{(4n+1)p}C_{4M}^{(4m+3)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x(N-1-n,M-1-m)C_{4N}^{(4n+3)p}C_{4M}^{(4m+3)q} . \end{aligned} \tag{6}$$

which enables to change the summing limits:

$$\begin{aligned} L_T^{II}(p,q) = \sum_{n=0}^{N/2-1}\sum_{m=0}^{M/2-1} x(n,m)C_{4N}^{(4n+1)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=N/2}^{N-1}\sum_{m=0}^{M/2-1} x(n,m)C_{4N}^{(4(N-n)-1)p}C_{4M}^{(4m+1)q} + \\ \sum_{n=0}^{N/2-1}\sum_{m=M/2}^{M-1} x(n,m)C_{4N}^{(4n+1)p}C_{4M}^{(4(M-m)-1)q} + \\ \sum_{n=N/2}^{N-1}\sum_{m=M/2}^{M-1} x(n,m)C_{4N}^{(4(N-n)-1)p}C_{4M}^{(4(M-m)-1)q} . \end{aligned} \tag{7}$$

Using the following identity for p and for q, similarly:

$$C_{4N}^{(4(N-n)-1)p} = \cos\left(2\pi\frac{(4n+1)\,p}{4N} - 2\pi p\right) = C_{4N}^{(4n+1)p} \;, \tag{8}$$

we can sum all the elements in eq. (7) directly, finally arriving at:

$$L_T^{II}(p,q) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m) C_{4N}^{(4n+1)p} C_{4M}^{(4m+1)q} \;. \tag{9}$$

Let us now consider computing the equation (9) in four cases. Considering the identities:

$$\begin{aligned} C_{4N}^{(4n+1)p} &= C_N^{np} C_{4N}^{p} - S_N^{np} S_{4N}^{p} \;, \\ C_{4N}^{(4n+1)(N-p)} &= S_N^{np} C_{4N}^{p} + C_N^{np} S_{4N}^{p} \;, \end{aligned} \tag{10}$$

and for q, similarly, we can write:

$$\begin{aligned} L_T^{II}(p,q) &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m) \left(C_N^{pn} C_{4N}^{p} - S_N^{pn} S_{4N}^{p}\right)\left(C_M^{qm} C_{4M}^{q} - S_M^{qm} S_{4M}^{q}\right) , \\ L_T^{II}(p,M-q) &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m) \left(C_N^{pn} C_{4N}^{p} - S_N^{pn} S_{4N}^{p}\right)\left(S_M^{qm} C_{4M}^{q} + C_M^{qm} S_{4M}^{q}\right) , \\ L_T^{II}(N-p,q) &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m) \left(S_N^{pn} C_{4N}^{p} + C_N^{pn} S_{4N}^{p}\right)\left(C_M^{qm} C_{4M}^{q} - S_M^{qm} S_{4M}^{q}\right) , \\ L_T^{II}\;(N-p,\,M-q) &= \\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m) \left(S_N^{pn} C_{4N}^{p} + C_N^{pn} S_{4N}^{p}\right)\left(S_M^{qm} C_{4M}^{q} + C_M^{qm} S_{4M}^{q}\right) , \end{aligned} \tag{11}$$

where $p = 1, ..., N/2-1, q = 1, ..., M/2-1$.

Therefore, taking into account the equations (2) we finally arrive at:

$$\begin{aligned} Re\{X_{N\times M}(p,q)\} &= L_T^{II}(p,q) C_{4NM}^{pM+qN} + L_T^{II}(p,M-q) S_{4NM}^{pM+qN} + \\ &\quad L_T^{II}(N-p,q) S_{4NM}^{pM+qN} - L_T^{II}(N-p,M-q) C_{4NM}^{pM+qN} \;, \\ Im\{X_{N\times M}(p,q)\} &= L_T^{II}(p,q) S_{4NM}^{pM+qN} - L_T^{II}(p,M-q) C_{4NM}^{pM+qN} - \\ &\quad L_T^{II}(N-p,q) C_{4NM}^{pM+qN} - L_T^{II}(N-p,M-q) S_{4NM}^{pM+qN} \;, \\ Re\{X_{N\times M}(N-p,q)\} &= L_T^{II}(p,q) C_{4NM}^{pM-qN} - L_T^{II}(p,M-q) S_{4NM}^{pM-qN} + \\ &\quad L_T^{II}(N-p,q) S_{4NM}^{pM-qN} + L_T^{II}(N-p,M-q) C_{4NM}^{pM-qN} \;, \\ Im\{X_{N\times M}(N-p,q)\} &= -L_T^{II}(p,q) S_{4NM}^{pM-qN} - L_T^{II}(p,M-q) C_{4NM}^{pM-qN} + \\ &\quad L_T^{II}(N-p,q) C_{4NM}^{pM-qN} - L_T^{II}(N-p,M-q) S_{4NM}^{pM-qN} \;. \end{aligned} \tag{12}$$

It is worth noting that there is no need of performing separate computations for $X_{N\times M}(p, M-q)$ and $X_{N\times M}(N-p, M-q)$, due to the Fourier symmetry property for real signals. Also, the computations may be simplified for $p=0$, $q=0$, $p=N/2$ and $q=M/2$.

Considering the graph of two-dimensional homogeneous, two-stage cosine transform, type II (2D FCT2, [4]) it can be seen that the equations (12) may be joined with the computations performed by two last layers of this graph. The transform which should be performed by these two layers after necessary modification is given as:

$$\begin{bmatrix} Re\{X_{N\times M}(p,q)\} \\ Im\{X_{N\times M}(p,q)\} \\ Re\{X_{N\times M}(N-p,q)\} \\ Im\{X_{N\times M}(N-p,q)\} \end{bmatrix} = A \cdot \begin{bmatrix} L_{T1}(p,q) \\ L_{T2}(p, M/2-q) \\ L_{T3}(N/2-p, q) \\ L_{T4}(N/2-p, M/2-q) \end{bmatrix} , \qquad (13)$$

where the matrix A is defined as follows:

$$A = \begin{bmatrix} C_{4NM}^{2(pM+qN)} & S_{4NM}^{2(pM+qN)} & S_{4NM}^{2(pM+qN)} & -C_{4NM}^{2(pM+qN)} \\ S_{4NM}^{2(pM+qN)} & -C_{4NM}^{2(pM+qN)} & -C_{4NM}^{2(pM+qN)} & -S_{4NM}^{2(pM+qN)} \\ C_{4NM}^{2(pM-qN)} & -S_{4NM}^{2(pM-qN)} & S_{4NM}^{2(pM-qN)} & C_{4NM}^{2(pM-qN)} \\ -S_{4NM}^{2(pM-qN)} & -C_{4NM}^{2(pM-qN)} & C_{4NM}^{2(pM-qN)} & -S_{4NM}^{2(pM-qN)} \end{bmatrix} , \qquad (14)$$

and $L_{T1}, ...L_{T4}$ denote subblocks $L_1, ...L_4$ (cf. the 2D FCT2 graph construction in [4]) performing $DCT^{II}_{\frac{N}{2}\times\frac{M}{2}}$ transform of the signal permuted with formulae (4).

The matrix A must be factorized in order to be computed by the two layers. Preferably, their connection scheme should be the same as in the (2D FCT2) algorithm to preserve the homogeneous structure of the whole graph. As every row of the matrix A contain only two distinct coefficients (ignoring the sign), its natural factorization yields only trivial operations (additions/subtractions) in the one-but-last layer. The problem is that these operations affect the pair of blocks: L_{T1}, L_{T4} and the pair: L_{T2}, L_{T3}, which violates the homogeneous scheme of the one-but-last layer, according to which the operations should affect the pairs: L_{T1}, L_{T2} and L_{T3}, L_{T4}.

This difficulty may be solved by swapping the blocks L_{T2}, L_{T4}. As their structure and the coefficients of their basic operations are practically the same (except of the last layer which needs minor adjustment) it is enough to simply swap their inputs. This may be performed by an additional input permutation:

$$R(n) = \begin{cases} x(n) & \text{, for } n < N/4 \text{ ;} \\ x(n+N/4) & \text{, for } n = N/4, N/4+1, ..., N/2-1 \text{ ;} \\ x(n-N/4) & \text{, for } n = N/2, N/2+1, ..., 3\cdot N/4-1 \text{ ;} \\ x(n) & \text{, for } n \geq 3\cdot N/4 \text{ .} \end{cases}$$

The permutation R is performed directly before the first layer of basic operations of the whole graph. It should be noted that swapping the blocks L_{T2}, L_{T4} implies some changes in the types of the basic operations in the last two layers (cf. [4])

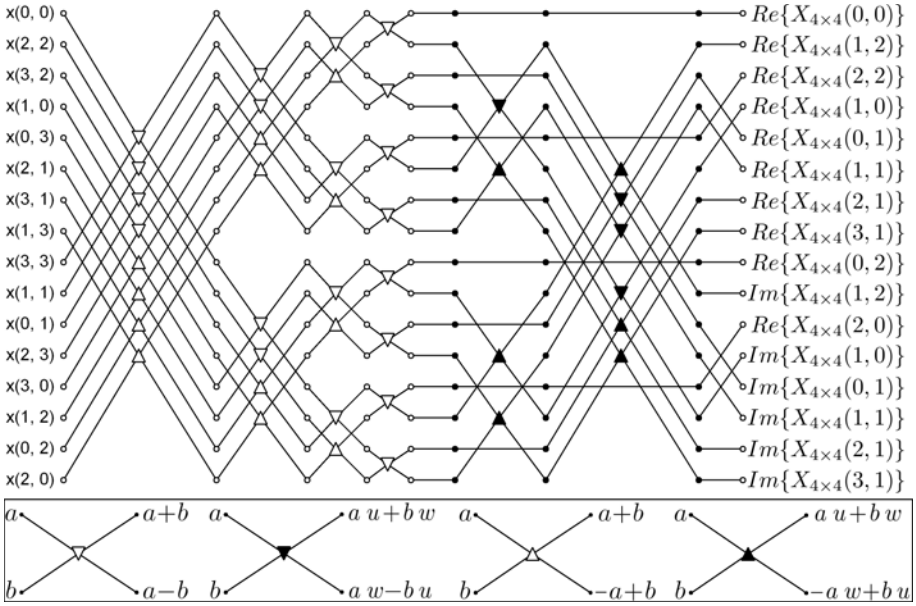

x(0, 0)
x(2, 2)
x(3, 2)
x(1, 0)
x(0, 3)
x(2, 1)
x(3, 1)
x(1, 3)
x(3, 3)
x(1, 1)
x(0, 1)
x(2, 3)
x(3, 0)
x(1, 2)
x(0, 2)
x(2, 0)
$Re\{X_{4\times4}(0,0)\}$
$Re\{X_{4\times4}(1,2)\}$
$Re\{X_{4\times4}(2,2)\}$
$Re\{X_{4\times4}(1,0)\}$
$Re\{X_{4\times4}(0,1)\}$
$Re\{X_{4\times4}(1,1)\}$
$Re\{X_{4\times4}(2,1)\}$
$Re\{X_{4\times4}(3,1)\}$
$Re\{X_{4\times4}(0,2)\}$
$Im\{X_{4\times4}(1,2)\}$
$Re\{X_{4\times4}(2,0)\}$
$Im\{X_{4\times4}(1,0)\}$
$Im\{X_{4\times4}(0,1)\}$
$Im\{X_{4\times4}(1,1)\}$
$Im\{X_{4\times4}(2,1)\}$
$Im\{X_{4\times4}(3,1)\}$
a
b
$a+b$
$a-b$
a
b
$a\,u+b\,w$
$a\,w-b\,u$
a
b
$a+b$
$-a+b$
a
b
$a\,u+b\,w$
$-a\,w+b\,u$

Fig. 1. The graph of the network for Fourier transform computation (size: 4×4)

for the cases when $p = 0$, $q = 0$, $p = N/2$ and $q = M/2$. The complete graph of the 4×4 network is presented in Fig. 1. This graph may be used either for direct Fourier transform computation or, alternatively, its basic operations may be treated as BOONs [3], where the coefficients u, w are neural weights adapted by any gradient optimization method. Note that applying a special algorithm with tangent multipliers enables to reduce the number of weights from two to one per each BOON [3].

3 Amplitude Spectrum of Two-Dimensional Fourier Transform

Similarly to the one-dimensional case [6], adding a special output layer is necessary if we want the network to be able to compute Fourier amplitude spectrum (Fig. 2). Its connection scheme is quite simple, as its every basic operation receives, as inputs, the real and the imaginary part of the same spectrum element, respectively. Hence the obtained sequence of the amplitude spectrum elements, with an exception of the last element, is the same as in the case of the real spectrum.

It should be stressed that this is the only non-linear layer in the network. It is also worth noting that the set of linear transforms possible to obtain with the preceding layers is considerably confined due to their reduced connection scheme.

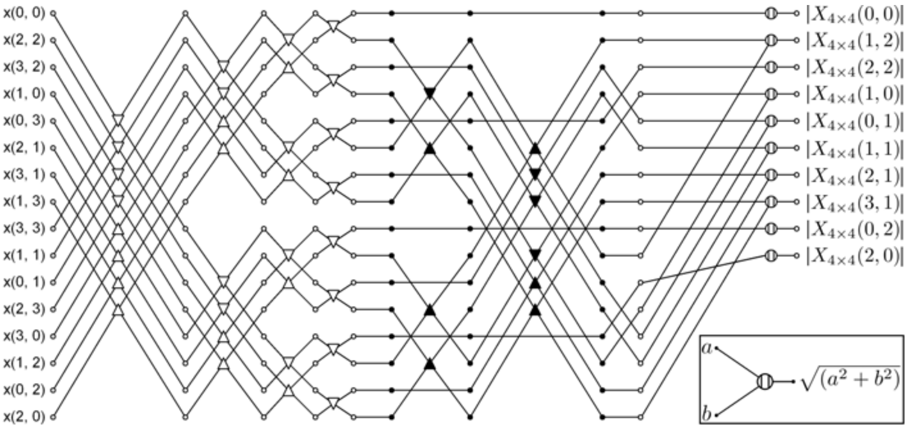


Fig. 2. The graph of the network for amplitude spectrum computation (size: 4 × 4)

4 Experimental Validation

Image recognition experiments have been performed on the example of ETH-80 database [7], using the testing scheme proposed by its authors (leave-one-out crossvalidation), with two different neural networks. The first one (FONN) was the network from Fig. 2 appended with a single linear output layer. The second network was a classical perceptron with one hidden layer comprised of non-linear neurons with sigmoidal, uni-polar activation function (MLP). Both networks were tested under the same conditions; in particular the same gradient optimization methods were applied with minimal validation error stopping criterion. In both cases the preprocessing was confined to downsampling the images to $N \times M = 32 \times 32$, normalization and constant component removal. Several different numbers of hidden neurons K were used for the MLP (denoted as MLPK). The recognition results are presented in Table 1.

The superiority of the proposed sparse neural architecture over the typical "dense" multilayer network is clearly visible not only in the significantly better

Table 1. Image recognition results

	MLP8	MLP16	MLP32	MLP64	FONN	FONN
$N \times M$	32×32	32×32	32×32	32×32	32×32	64×64
Number of weights	8,272	16,536	33,064	66,120	8,728	38,936
Recognition rate	66.59%	68.05%	69.85%	66.86%	76.77%	80.98%

Recognition per class (FONN, $N \times N = 64 \times 64$)

apple	car	cow	cup	dog	horse	pear	tomato
76.34%	97.32%	62.20%	99.02%	67.32%	64.88%	99.76%	80.98%

recognition rates but also in the reduced number of weights to adapt. FONN application enables to avoid the problem of searching for the optimal number of hidden neurons. The last column of Table 1 shows that it may be better to increase the input space dimensionality and use the FONN instead of increasing the number of hidden neurons in MLP which, for the examined problem, may easily lead to slowing down the training process with no positive effect on the classification outcome. Note that in all the presented cases the recognition rate on the *training* set was in the range of 95.34% - 98.43% so the obtained results may be interpreted in terms of the generalization error. Moreover, the process of training the FONN is stable and converges faster than for the MLP. Usually the same level of training error was obtained after half or one-third of the number of epochs needed by the MLP. Searching for the best result on the test set obtained anytime during the training yielded 85.85% recognition for the FONN from the last column of the Table 1. This value, which may be seen as the upper possible limit of the classifier capabilities is comparable to the best results reported in [7].

5 Conclusion

Neural network with architecture based on fast two-stage algorithm of Fourier transform have been constructed via modifications of cosine transform algorithm. The network, appended with the amplitude-computing layer and output linear layer, proved to be suitable for raw image analysis and classification, providing good recognition rate and low computational complexity. The comparison with multilayer perceptron showed the enhanced performance, stability and faster convergence of the training process offered by the proposed solution.

This work was supported by the Polish Ministry of Science and Higher Education (grant no. N N516 4308 33).

References

1. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. of the IEEE 86(11), 2278–2324 (1998)
2. Tao, D., Li, X., Wu, X., Maybank, S.J.: Tensor Rank One Discriminant Analysis - A convergent method for discriminative multilinear subspace selection. Neurocomputing 71(10-12), 1866–1882 (2008)
3. Stasiak, B., Yatsymirskyy, M.: Fast Orthogonal Neural Networks. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 142–149. Springer, Heidelberg (2006)
4. Stasiak, B., Yatsymirskyy, M.: Fast homogeneous algorithm of two-dimensional cosine transform, type II with tangent multipliers. Electrotechnical Review 12/2008, pp. 290–292 (2008)
5. Rao, K.R., Yip, P.: Discrete cosine transform. Academic Press, San Diego (1990)
6. Stasiak, B., Yatsymirskyy, M.: Fast orthogonal neural network for adaptive Fourier amplitude spectrum computation in classification problems. In: ICMMI (2009)
7. Leibe, B., Schiele, B.: Analyzing Appearance and Contour Based Methods for Object Categorization. In: Proc. of the International Conf. CVPR 2003, pp. 409–415 (2003)

An Enhanced Probabilistic Neural Network Approach Applied to Text Classification

Patrick Marques Ciarelli and Elias Oliveira

Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514
29075-910, Vitória - ES, Brazil
{pciarelli,elias}@lcad.inf.ufes.br

Abstract. Text classification is still a quite difficult problem to be dealt with both by the academia and by the industrial areas. On the top of that, the importance of aggregating a set of related amount of text documents is steadily growing in importance these days. The presence of multi-labeled texts and great quantity of classes turn this problem even more challenging. In this article we present an enhanced version of Probabilistic Neural Network using centroids to tackle the multi-label classification problem. We carried out some experiments comparing our proposed classifier against the other well known classifiers in the literature which were specially designed to treat this type of problem. By the achieved results, we observed that our novel approach were superior to the other classifiers and faster than the Probabilistic Neural Network without the use of centroids.

Keywords: Information Retrieval, Probabilistic Neural Network, Multi-label Problem.

1 Introduction

Automatic text classification is an activity that is becoming more and more important nowadays. This might be due to the huge amount of information available and the great challenge for the information retrieval. In addition, many of real databases are multi-labeled and have a great amount of categories, which make the text classification task even more difficult [1]. Such problems are tackled by the information retrieval (IR) communities, both in academic and industrial contexts.

To treat such issues, in this paper we used a slightly modified version of the standard structure of the Probabilistic Neural Network (PNN) presented in [3]. In this modified version, we used centroids for the training of the PNN. In order to evaluate these PNN's versions, the classical one and that proposed by us in this paper, we used a set of multi-labeled Yahoo's databases, initially used in [6]. Furthermore, we compared their results against the other specialized classifiers in multi-labeled classification. Both versions of the PNN achieved better results, in our evaluation, than that performed by the other classifiers. The enhanced PNN with centroids was the best in our evaluation.

This work is organized as follows. In Section 2, we detail our algorithms. We describe the metrics used to evaluate in Section 3. In Section 4, the experiments and results are discussed. Finally, we present our conclusions in Section 5.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 661–668, 2009.

2 The Algorithms

The Probabilistic Neural Network is an artificial neural network for nonlinear computing which approaches the Bayes optimal decision boundaries. The original PNN algorithm [2] was designed for single-label problems. Thus, its standard architecture was slightly modified, so that it is now capable of solving multi-labeled problems.

In this modified version, instead of four, the PNN is composed of only three layers: the *input* layer, the *pattern* layer and the *summation* layer, as it is showed in Figure 1. Thus, like in the original structure, this version of PNN needs only one training step, therefore, its training is faster than other well known feed-forward neural networks [4].

The training consists in assigning each training sample w_j of category C_j to a neuron of pattern layer of category C_j. Thus, the weight vector of this neuron is the characteristics vector of the sample.

For each d_j test instance passed by the input layer to a neuron in the pattern layer, it computes the output for the d_j. The computation is showed in Equation 1.

$$F_{k,i}(d_j) = \frac{1}{2\pi\sigma^2} exp(\frac{d_j^t w_{k,i} - 1}{\sigma^2}), \tag{1}$$

where the d_j is the pattern characteristics input vector, and the $w_{k,i}$ is the k^{th} sample for a neuron of category C_i, $k \in N_i$, whereas N_i is the number of neurons of C_i. In addition, d_j and $w_{k,i}$ were normalized so that $d_j^t d_j = 1$ and $w_{k,i}^t w_{k,i} = 1$. The σ is the Gaussian standard deviation, which determines the receptive field of the Gaussian curve.

The next step is the summation layer. In this layer, all outputs of the previous layer are summed, Equation 2, in each cluster C_i producing $p_i(d_j)$ values, where $|C|$ is the total number of categories and h_i is the priori probability of the class C_i. Whether we consider the priori probability from database of training, so we can inconsiderate the fraction $\frac{h_i}{N_i}$.

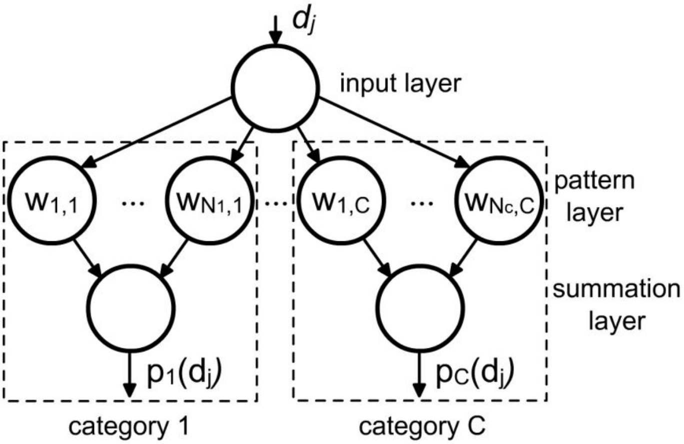


Fig. 1. The modified Probabilistic Neural Network architecture

$$p_i(d_j) = \frac{h_i}{N_i} \sum_{k=1}^{N_i} F_{k,i}(d_j) \quad i = 1,2,\ldots,|C| \,. \tag{2}$$

Finally, for the selection of the categories which will be assigned by the neural network to each sample, we consider the most likely categories pointed out by the summation layer based on a chosen threshold.

Differently from other types of networks, such as those feed forward based, the proposed PNN needs few parameters to be configured: the σ, (see in Equation 1) and the determination of threshold value. Another advantage of the probabilistic neural networks is that it is easy to add new categories, or new training inputs, into the already running structure, which is good for the on-line applications [4]. On the other hand, one of its drawbacks is the great number of neurons in the pattern layer that, as a consequence, may produce a high consumption of memory and slow rate of classification. Moreover, the presence of repeated samples may harm the performance of the classifier.

2.1 Probabilistic Neural Network with Centroids

To minimize the drawbacks of the PNN, we propose a technique of centroids, such that is used only one neuron for class in the pattern layer. Equation 3 shows a mathematical procedure to obtain the centroid for each class C_i, where $w_{k,i}$ is the k^{th} sample of training of class C_i, N_i is the number of samples of C_i and W_i is the obtained centroid. To reduce the loss of information, we also obtain from database the priori probability h_i of each class. Thus, the fraction $\frac{h_i}{N_i}$ of Equation 2 will be reduced to h_i, because N_i will be equal to 1, and the centroid W_i is associated to the neuron of pattern layer of class C_i. Hence, with this procedure, the PNN will have only one neuron per category in the pattern layer.

$$W_i = \frac{1}{N_i} \sum_{k=1}^{N_i} w_{k,i} \quad h_i = N_i \quad i = 1,\ldots,|C| \,. \tag{3}$$

3 Metrics

Formalizing the problem we have at hand, text categorization may be defined as the task of assigning documents to a predefined set of categories, or classes [1]. In multi-label text categorization a document may be assigned to one or more categories. Let $\mathcal{D}$ be the domain of documents, $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ a set of pre-defined categories, and $\Omega = \{d_1, d_2, \ldots, d_{|\Omega|}\}$ an initial corpus of documents previously categorized by some human specialists into subsets of categories of $\mathcal{C}$.

In multi-label learning, the training(-and-validation) set $TV = \{d_1, d_2, \ldots, d_{|TV|}\}$ is composed of a number documents, each associated with a subset of categories in $\mathcal{C}$. TV is used to train and validate (actually, to tune eventual parameters of) a categorization system that associates the appropriate combination of categories to the characteristics of each document in the TV. The test set $Te = \{d_{|TV|+1}, d_{|TV|+2}, \ldots, d_{|\Omega|}\}$, on the other hand, consists of documents for which the categories are unknown to the automatic categorization systems. TV has $|TV|$ samples and Te has $|\Omega| - |TV| = p$ samples. After being trained, as well as tuned, by the TV, the categorization systems are used to predict the set of categories of each document in Te.

A multi-label categorization system typically implements a real-valued function of the form $f : \mathcal{D} \times \mathcal{C} \to \mathbb{R}$ that returns a value for each pair $\langle d_j, c_j \rangle \in \mathcal{D} \times \mathcal{C}$ that, roughly speaking, represents the evidence for the fact that the test document d_j should be categorized under the category $c_j \in C_j$, where $C_j \subset \mathcal{C}$. The real-valued function $f(.,.)$ can be transformed into a ranking function $r(.,.)$, which is an one-to-one mapping onto $\{1, 2, \ldots, |\mathcal{C}|\}$ such that, if $f(d_j, c_1) > f(d_j, c_2)$, then $r(d_j, c_1) < r(d_j, c_2)$. If C_j is the set of proper categories for the test document d_j, then a successful categorization system tends to rank categories in C_j higher than those not in C_j. Additionally, we also use a threshold parameter so that those categories that are ranked above the threshold τ (*i.e.*, $c_k | f(d_j, c_k) \geq \tau$) are the only ones to be assigned to the test document.

We have used five multi-label metrics discussed in [6] to evaluate the performance of classifiers. We now present each one of these metrics:

Hamming Loss (hloss) evaluates how many times the test document d_j is misclassified, *i.e.*, a category not belonging to the document is predicted or a category belonging to the document is not predicted.

$$\text{hloss} = \frac{1}{p} \sum_{j=1}^{p} \frac{1}{|C|} |P_j \Delta C_j|, \tag{4}$$

where $|C|$ is the number of categories and Δ is the symmetric difference between the set of predicted categories P_j and the set of appropriate categories C_j of the test document d_j. The predicted categories are those which rank higher than the threshold τ.

One-error (one-error) evaluates if the top ranked category is present in the set of appropriate categories C_j of the test document d_j.

$$\text{one-error} = \frac{1}{p} \sum_{j=1}^{p} \text{error}_j, \quad \text{error}_j = \begin{cases} 0 \text{ if } [\arg\max_{c \in \mathcal{C}} f(d_j, c)] \in C_j \\ 1 \text{ otherwise.} \end{cases} \tag{5}$$

where $[\arg\max_{c \in \mathcal{C}} f(d_j, c)]$ returns the top ranked category for the test document d_j.

Coverage (coverage) measures how far we need to go down the rank of categories in order to cover all the possible categories assigned to a test document.

$$\text{coverage} = \frac{1}{p} \sum_{j=1}^{p} (\max_{c \in C_j} r(d_j, c) - 1), \tag{6}$$

where $\max_{c \in C_j} r(d_j, c)$ returns the maximum rank for the set of appropriate categories of the test document d_j.

Ranking Loss (rloss) evaluates the fraction of category pairs $\langle c_k, c_l \rangle$, for which $c_k \in C_j$ and $c_l \in \bar{C}_j$, that are reversely ordered for the test document d_j:

$$\text{rloss} = \frac{1}{p} \sum_{j=1}^{p} \frac{|\{(c_k, c_l) | f(d_j, c_k) \leq f(d_j, c_l)\}|}{|C_j||\bar{C}_j|}, \tag{7}$$

where $(c_k, c_l) \in C_j \times \bar{C}_j$, and $\bar{C}_j$ is the complementary set of C_j in $\mathcal{C}$.

Average Precision (avgprec) evaluates the average of precision computed after truncating the ranking of categories after each category $c_i \in C_j$ in turn:

$$\text{avgprec} = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{|C_j|}\sum_{k=1}^{|C_j|} precision_j(R_{jk}), \tag{8}$$

where R_{jk} is the set of ranked categories that goes from the top ranked category until a ranking position k, where there is a category $c_i \in C_j$ for d_j, and $precision_j(R_{jk})$ is the number of pertinent categories in R_{jk} divided by $|R_{jk}|$.

The smaller the value of *hamming loss*, *one-error*, *coverage* and *ranking loss*, and the larger the value of *average precision*, the better the performance of the categorization system. The performance is optimal when hloss = one-error = rloss = 0 and avgprec = 1.

4 Experiments

We carry out a series of experiments to compare the versions of PNN against the classifiers: ML-kNN, that is based on the kNN [6], Rank-SVM [9], a modified version of SVM, ADTBoost.MH [7] and BoosTexter [8], that both are techniques based on decision trees. We have used 11 text databases from Yahoo domain in our experiments[1], where each database has 2000 samples to training and 3000 to test, the average number of classes is 30 and there is a mean of 1.48 classes assigned by sample. To evaluate the performance of the algorithms we used the metrics presented in Section 3[2], and the results were obtained directly from the [6], with exception of the PNNs' results.

In [6] is not mentioned any use of a search strategy for the optimization of the classifiers' parameters. To turn it in a fair comparison with the other techniques, we will test our approaches of PNNs considering only the order of magnitude of the variance's value. For this, we used part of training set of Arts database from Yahoo and we tested the variance's values 10, 1 and 0.1 on a cross-validation experiment. The chosen value was 0.1. The threshold's value used to the Hamming Loss metric was 0.5, the same value used by ML-kNN, therefore, this parameter was also not optimized to the PNNs.

The results yielded with the use of the Yahoo's database are presented in Tables from 1 to 5. Each one of the tables represents a metric, where each row is a data set and each column is a classifier. The term "Average" in the last row means the average value of the metric obtained by each classifier to all databases.

To accomplish a clearer evaluation of the classifiers, we adopted two criteria derived from [6]. The first criterion creates one partial order " $\succ$ " that evaluates the performance between two classifiers for each metric. In that way, if the classifier $A1$ has a better performance than $A2$ to a given metric, so we have $A1 \succ A2$. In order to perform this task, we used two-tailed paired t-test at 5% significance level.

However, the presented criterion is insufficient to obtain the performance of classifiers as a whole, therefore, we used a second criterion. In this one is applied a system

[1] Databases and codes of the versions of PNN are available at http://www.inf.ufes.br/~elias/ciarp2009.zip

[2] Ranking Loss to ADTBoost.MH was not reported because, according to [6], the algorithm of this classifier did not supply such information.

Table 1. Hamming Loss obtained by the classifiers

Data Set	ML-kNN	BoosTexter	ADTBoost.MH	Rank-SVM	PNN	PNN-centroid
Arts&Humanities	0.0612	0.0652	**0.0585**	0.0615	0.0630	0.0626
Business&Economy	**0.0269**	0.0293	0.0279	0.0275	0.0307	0.0289
Computers&Internet	0.0412	0.0408	0.0396	**0.0392**	0.0447	0.0412
Education	**0.0387**	0.0457	0.0423	0.0398	0.0437	0.0437
Entertainment	0.0604	0.0626	**0.0578**	0.0630	0.0640	0.0635
Health	0.0458	**0.0397**	**0.0397**	0.0423	0.0514	0.0481
Recreation&Sports	0.0620	0.0657	**0.0584**	0.0605	0.0634	0.0631
Reference	0.0314	0.0304	0.0293	0.0300	0.0307	**0.0289**
Science	**0.0325**	0.0379	0.0344	0.0340	0.0353	0.0353
Social&Science	**0.0218**	0.0243	0.0234	0.0242	0.0281	0.0245
Society&Culture	**0.0537**	0.0628	0.0575	0.0555	0.0596	0.0599
Average	0.0432	0.0459	**0.0426**	0.0434	0.0468	0.0454

Table 2. One-Error obtained by the classifiers

Data Set	ML-kNN	BoosTexter	ADTBoost.MH	Rank-SVM	PNN	PNN-centroid
Arts&Humanities	0.6330	0.5550	0.5617	0.6653	0.5597	**0.5293**
Business&Economy	**0.1213**	0.1307	0.1337	0.1237	0.1317	0.1313
Computers&Internet	0.4357	0.4287	0.4613	**0.4037**	0.4457	0.4557
Education	0.5207	0.5587	0.5753	**0.4937**	0.5463	0.5420
Entertainment	0.5300	**0.4750**	0.4940	0.4933	0.5530	0.4960
Health	0.4190	**0.3210**	0.3470	0.3323	0.4080	0.3807
Recreation&Sports	0.7057	0.5557	**0.5547**	0.5627	0.6037	0.5670
Reference	0.4730	0.4427	0.4840	**0.4323**	0.4780	0.4727
Science	0.5810	0.6100	0.6170	**0.5523**	0.6123	0.5930
Social&Science	**0.3270**	0.3437	0.3600	0.3550	0.3753	0.3703
Society&Culture	0.4357	0.4877	0.4845	**0.4270**	0.4647	0.4637
Average	0.4711	0.4463	0.4612	**0.4401**	0.4708	0.4547

Table 3. Coverage obtained by the classifiers

Data Set	ML-kNN	BoosTexter	ADTBoost.MH	Rank-SVM	PNN	PNN-centroid
Arts&Humanities	5.4313	5.2973	5.1900	9.2723	4.8503	**4.6250**
Business&Economy	2.1840	2.4123	2.4730	3.3637	2.1087	**2.0527**
Computers&Internet	4.4117	4.4887	4.4747	8.7910	4.0380	**3.8963**
Education	3.4973	4.0673	3.9663	8.9560	3.4980	**3.4067**
Entertainment	3.1467	3.0883	3.0877	6.5210	3.0663	**2.8883**
Health	3.3043	3.0780	3.0843	5.5400	3.0093	**2.8730**
Recreation&Sports	5.1010	4.4737	4.3380	5.6680	4.2773	**4.0573**
Reference	3.5420	3.2100	3.2643	6.9683	2.9097	**2.7560**
Science	6.0470	6.6907	6.6027	12.401	5.9930	**5.6180**
Social&Science	3.0340	3.6870	3.4820	8.2177	3.1357	**2.9430**
Society&Culture	5.3653	5.8463	**4.9545**	6.8837	5.3350	5.2323
Average	4.0968	4.2127	4.0834	7.5075	3.8383	**3.6681**

Table 4. Ranking Loss obtained by the classifiers

Data Set	ML-kNN	BoosTexter	ADTBoost.MH	Rank-SVM	PNN	PNN-centroid
Arts&Humanities	0.1514	0.1458	N/A	0.2826	0.1306	**0.1223**
Business&Economy	0.0373	0.0416	N/A	0.0662	0.0367	**0.0349**
Computers&Internet	0.0921	0.0950	N/A	0.2091	0.0826	**0.0787**
Education	0.0800	0.0938	N/A	0.2080	0.0803	**0.0773**
Entertainment	0.1151	0.1132	N/A	0.2617	0.1103	**0.1025**
Health	0.0605	0.0521	N/A	0.1096	0.0526	**0.0491**
Recreation&Sports	0.1913	0.1599	N/A	0.2094	0.1556	**0.1432**
Reference	0.0919	0.0811	N/A	0.1818	0.0732	**0.0685**
Science	0.1167	0.1312	N/A	0.2570	0.1166	**0.1073**
Social&Science	0.0561	0.0684	N/A	0.1661	0.0601	**0.0546**
Society&Culture	0.1338	0.1483	N/A	0.1716	0.1315	**0.1286**
Average	0.1024	0.1028	N/A	0.1930	0.0936	**0.0879**

Table 5. Average Precision obtained by the classifiers

Data Set	ML-kNN	BoosTexter	ADTBoost.MH	Rank-SVM	PNN	PNN-centroid
Arts&Humanities	0.5097	0.5448	0.5526	0.4170	0.5645	**0.5851**
Business&Economy	**0.8798**	0.8697	0.8702	0.8694	0.8763	0.8779
Computers&Internet	0.6338	**0.6449**	0.6235	0.6123	0.6398	0.6420
Education	**0.5993**	0.5654	0.5619	0.5702	0.5889	0.5980
Entertainment	0.6013	**0.6368**	0.6221	0.5637	0.5991	0.6295
Health	0.6817	**0.7408**	0.7257	0.6839	0.7047	0.7207
Recreation&Sports	0.4552	0.5572	0.5639	0.5315	0.5396	**0.5672**
Reference	0.6194	**0.6578**	0.6264	0.6176	0.6441	0.6512
Science	**0.5324**	0.5006	0.4940	0.5007	0.5073	0.5278
Social&Science	**0.7481**	0.7262	0.7217	0.6788	0.7113	0.7272
Society&Culture	**0.6128**	0.5717	0.5881	0.5717	0.5993	0.6018
Average	0.6249	0.6378	0.6318	0.6015	0.6341	**0.6480**

Table 6. Relative performance of the classifiers by the two criteria

HL-Hamming Loss; OE-One-error; C-Coverage; RL-Ranking Loss; AP-Average Precision A1-ML-kNN; A2-BoosTexter; A3-ADTBoost.MH; A4-Rank-SVM; A5-PNN; A6-PNN-centroid	
Metrics	Criterion 1
HL	$A1 \succ A5, A1 \succ A6, A3 \succ A2, A4 \succ A2, A3 \succ A5, A3 \succ A6, A4 \succ A5, A4 \succ A6, A6 \succ A5$
OE	$A2 \succ A3, A2 \succ A5, A6 \succ A5$
C	$A1 \succ A4, A5 \succ A1, A6 \succ A1, A2 \succ A4, A5 \succ A2, A6 \succ A2, A3 \succ A4, A5 \succ A3, A6 \succ A3, A5 \succ A4, A6 \succ A4, A6 \succ A5$
RL	$A1 \succ A4, A5 \succ A1, A6 \succ A1, A2 \succ A4, A5 \succ A2, A6 \succ A2, A5 \succ A4, A6 \succ A4, A6 \succ A5$
AP	$A2 \succ A4, A3 \succ A4, A6 \succ A3, A5 \succ A4, A6 \succ A4, A6 \succ A5$
Techniques	Criterion 2
ML-kNN	{PNN, PNN-centroid} > **ML-kNN** > Rank-SVM
BoosTexter	{PNN,PNN-centroid} > **BoosTexter** > Rank-SVM
ADTBoost.MH	PNN-centroid > **ADTBoost.MH** > Rank-SVM
Rank-SVM	{ML-kNN, BoosTexter, ADTBoost.MH,PNN, PNN-centroid} > **Rank-SVM**
PNN	PNN-centroid > **PNN** > {ML-kNN, BoosTexter, Rank-SVM}
PNN-centroid	**PNN-centroid** > {ML-kNN, BoosTexter, Rank-SVM, ADTBoost.MH, PNN}

based on rewards and punishes. For example, for the case of $A1 \succ A2$ the classifier $A1$ is rewarded with +1 and the classifier $A2$ is punished with -1. Then, we compare the classifiers two a two through of the sum of their rewarded and punished between them. In this case, if $A1$ have a positive value in relation to $A2$, so $A1$ is superior to $A2$, *i. e.*, $A1 > A2$. Thus, the results obtained by the two criteria are shown in Table 6.

Table 6 shows that PNN-centroid is superior over to every algorithm, while the Rank-SVM is the worst. Moreover, the PNN shows to be the second better, being inferior just to the PNN-centroid and it had similar performance to ADTBoost.MH, whereas the other classifiers (ML-kNN, ADTBoost.MH and BoosTexter) were superior just to Rank-SVM. In addition, the PNN-centroid had a low time for classification (more than 10 times faster) and a small consumption of memory, when we compared with the PNN. Finally, the training phase of both PNNs was faster than the other algorithms.

5 Conclusions

The problem of text classification is still greatly challenging, due to the huge amount of information available. Other issues are the great quantity of classes and the presence of multi-labeled databases, which together increase the difficult of this task.

In this work, we presented an experimental evaluation on multi-label text classification of the performance of Probabilistic Neural Network and another version of it with centroids. Therefore, we conducted a comparative study of these PNNs and other four classifiers specially designed to solve this problem. The results showed that both versions of PNN devised by us presented good results, specially the PNN with centroids, that was superior to all the other classifiers. In addition, our approach is faster and consumed less memory than PNN without centroids.

A direction for future works is to study methods that can improve the results found in this article even more. Moreover, we are planning to tackle the problem of on-line learning using the proposed neural network in this paper.

Aknowledgement

This work was parcially supported by Receita Federal do Brasil and Fundação Espírito Santense de Tecnologia – FAPES-Brasil (grant 41936450/2008).

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
2. Specht, D.: Probabilistic Neural Networks, Oxford, UK, vol. 3(1), pp. 109–118. Elsevier Science Ltd., Amsterdam (1990)
3. Oliveira, E., Ciarelli, P.M., Badue, C., De Souza, A.F.: A Comparison between a KNN Based Approach and a PNN Algorithm for a Multi-label Classification Problem. In: ISDA 2008: Eighth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 628–633 (2008)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2001)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, 1st edn. Addison-Wesley, New York (1998)
6. Zhang, M.-L., Zhou, Z.-H.: MLkNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recognition 40(1), 2038–2048 (2007)
7. De Comité, F., Gilleron, R., Tommasi, M.: Learning Multi-label Alternating Decision Trees from Texts and Data. LNCS, pp. 35–49. Springer, Heidelberg (2003)
8. Schapire, R.E., Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization. Machine Learning, 135–168 (2000)
9. Elisseeff, A., Weston, J.: Kernel Methods for Multi-labelled Classification and Categorical Regression Problems. In: Advances in Neural Information Processing Systems, pp. 681–687 (2001)

Writer Identification Using Super Paramagnetic Clustering and Spatio Temporal Neural Network

Seyyed Ataollah Taghavi Sangdehi and Karim Faez

Qazvin Azad University of Iran, +98-911-155-5819
Amirkabir University, Tehran, Iran
atataghavi@gmail.com, kfaez@aut.ac.ir

Abstract. This paper discusses use of Super Paramagnetic Clustering (SPC) and Spatio Temporal Artificial Neuron in on-line writer identification, on Farsi handwriting. In online cases, speed and automation are advantages of one method on others, therefore we used unsupervised and relatively quick clustering method, which in comparison with conventional approaches, give us better result. Moreover, regardless of various parameters that available from acquisition systems, we only consider to displacement of pen tip at determined direction that lead to quick system due to its quick preprocessing and clustering. Also we use a threshold that remove displacement between disconnected point of a word that lead to a better classification result on on-line Farsi writers.

Keywords: Writer Identification, Super Paramagnetic Clustering, Spatio Temporal Neural Network.

1 Introduction

With the rapid development of the information technology, true user authentication, using biometrics information will be required to get the more reliable password. the necessity of person identification is increasing for example, in bank, shop, e-commerce and so on. In this work, person identification using handwriting, is referred to as writer identification. the target of writer identification is to quest for the personal identity, among a group of possible candidates, which is mainly used in security-oriented applications.

Writer identification problem is largely classified into two classes. One is offline methods, based on only static visual information, and the other is, on-line methods, based on dynamics of the handwriting process. A major advantage of the later method is very difficult to forge or copy the invisible dynamic features. We can say writer identification is identifying, writer from a written script such as character, signature and etc. Many methods of on-line signature recognition have been proposed at [1], [2], [3]. In the online methods, dynamic features of handwriting process have been used, such as pen point coordination [4], writing velocity [5], azimuth [6] and other features which are available from a digitizer. Due to the seemingly uniqueness of physiological and behavioral characteristics of each individual, writer identification has shown [1] to be a feasible task. Each writer's writing, has a set of characteristics which is exclusive to him, only.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 669–676, 2009.

However a few methods for on-line writer identification (exclusively, handwriting and not signature) have been presented. It is known that Farsi handwriting (words) consist of several stroks and are different from continuous English handwriting.

In this paper, we propose a novel approach for on-line writer identification based on Super Paramagnetic Clustering (SPC) algorithm [9] and Spatio Temporal Artificial Neuron (STAN) [10],[11] on Farsi handwriting. The rest of this paper is organized as follows. Section 2 gives a short overview of the SPC. Section 3 describes the writer identification procedure based on the SPC clustering and STAN classification. Experimental results and conclusion are provided in Section 4.

2 Overview of Super Paramagnetic Clustering

The key idea of Super Paramagnetic Clustering (SPC) is based on magnetic property of material at different temperature. Each material reach to high magnetic properties at special temperature. We use this temperature as a best value for clustering with SPC. The following is key ideas of Super Paramagnetic Clustering (SPC) [7], which is based on simulated interactions between each point and its k-nearest neighbors.

There are q different states per each magnetic particle. First step is to represent the m selected features of each spike i by a point $\boldsymbol{x}_i$ in an m-dimensional phase space. The interaction strength between points $\boldsymbol{x}_i$ is then defined as:

$$J_{ij} = \begin{cases} \frac{1}{k}\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2a^2}\right) & \text{If } \boldsymbol{x}_i \text{ is a nearest neighbor of } \boldsymbol{x}_j \\ 0 & \text{Else} \end{cases} \tag{1}$$

where a is the average nearest-neighbors distance and k is the number of nearest neighbors. Note that the strength of interaction J_{ij} between nearest neighbor spikes decays exponentially, with increasing Euclidean distance $d_{ij} = \left\|x_i - x_j\right\|^2$, which corresponds to the similarity of the selected features. In the second step, an initial random state s from 1 to q is assigned to each point $\boldsymbol{x}_i$ .Then N Monte Carlo iterations are run for different temperatures T, given an initial configuration of states s. A point $\boldsymbol{x}_i$ is randomly selected and its state s changed to a new state s_{new}, which is randomly chosen between 1 and q. probability that, the nearest neighbors of $\boldsymbol{x}_i$ will also change their state to s_{new} is given by:

$$P_{ij} = 1 - \exp\left(-\frac{J_{ij}}{T}\delta_{s_i,s_j}\right) \tag{2}$$

Where *T* is the temperature in which, this probability compute. Note that only those nearest neighbors of $\boldsymbol{x}_i$ which were in the same previous state *s,* are candidates to change their states to s_{new}. Neighbors which change their values, create a frontier and cannot change their state again, during the same iteration. Points which do not change their state in a first attempt, can do so, if revisited during the same iteration. Then for each point of the frontier, we apply equation (2), again to calculate the probability of changing state to s_{new} for their respective neighbors. The frontier is updated, and the update is repeated until the frontier does not change any more.

At that stage, we start the procedure again from another point and repeat it several times, in order to get the representative statistics. Points which are close together, (corresponding to a given cluster) change their state together. This can be quantified by measuring the point to point correlation δ_{s_i,s_j} and defining $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ to be members of the same cluster if $\delta_{s_i,s_j} \geq \theta$, for a given threshold θ. Clustering results, mainly depend on temperature and are robust to small changes on other parameters, like, threshold, number of nearest neighbors and states [9]. This method remains better results in sorting of spikes in comprise with other approach (Table.1).

3 Writer Identification Based on SPC and Spatio Temporal Neural Network

In this section we describe on-line writer identification. We get data from a tablet and apply preprocessing to produce displacement, which is converted to impulses in the form of spatio temporal coding in polar coordinates. Accumulated impulses at a temporal window go to a clustering unit as an input. After clustering, clustered data at another pass of algorithm, can produce impulses, like previous phase. Accumulated impulses from this section are representative of a person who must be identified.

3.1 Data Acquisition and Preprocessing

On-line writer identification methods, often use a data convertor device. The user registers his/her own written samples with a special pen, and handwriting, received online. The device we use at this experiment is a tablet from Wacom company, at A4 size, in which the sampling rate is 200 points per second. We record samples of each person in a text file. From the tablet we acquire signals of position coordinates of pen on surface of the tablet. Note that each handwriting word, consists of a sequence of pen tip coordinates. We can use displacement from these positions. Since the beginning of each word can be any point at writing surface, we use displacements at discrete directions [10], [11] which make this system translation invariant. Each displacement take place at one direction, by quantization of direction to nearest basic direction, among 8 defined direction which is the best number of features for input to our proposed method according to experiments. According to Fig.1, we make a vector with dimension equal to number of quantized direction that achieve at our experiment. Therefore we have 8 components. displacements at a direction take place on one of these components that can be an impulse at time. There are sequences of displacements at each handwriting. Then we must convert them to sequences of impulses.

This process is a spatio temporal problem. thus each displacement can be an impulse at a given time. We present spatio temporal coding by complex number [10]. Each signal x at time t is shown with amplitude η and phase ϕ (temporal position) from a reference time, at polar coordinate (η, ϕ)

$$x = \eta e^{i\phi} \text{ and } \tan(\phi) = \mu_t \tau \tag{3}$$

$$x = \eta e^{i \arctan(\mu_t \tau)} \tag{4}$$

and because of decreasing amplitude of x due to time, we have:

$$x_i(t) = \eta e^{-\mu_s \tau} e^{-i \arctan(\mu_t \tau)} \tag{5}$$

$$\mu_s = \mu_t = \frac{1}{TW} \tag{6}$$

$$x_j(t_2) = \eta_1 e^{-\mu_s (t_2 - t_1)} e^{-i \arctan(\mu_t (t_2 - t_1))} \tag{7}$$

Where x_i is an impulse at *i*th component of 8 dimensional displacement vector at time *t*, and TW is a temporal window in which feature vectors are created. When a new impulse x_j is emitted on a given component at time t_2 ,it is accumulated with the previous impulse according to (7). We need, creation of primary feature vector for clustering unit and also, feature vector for classification unit, so we consider the suitable temporal window which obtain from data set.

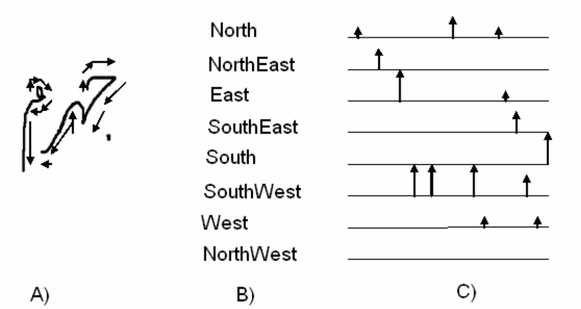


Fig. 1. A) Pen tip displacement at writing a word. B) Quantized direction. C) Sequence of displacement at quantized direction as impulse.

Preprocessing have an important role in obtaining better results. Preprocessing algorithm used in [11] was suitable for continuous English words. Since for Farsi handwritten word, each word can consist of one or some disjoint part, this algorithm can't work well on Farsi handwriting with disjoint sub words. Sudden jumping of pen tip at disjoint parts of a word that usually occurs between marks of a letter and other strokes of word or occurs between two disjoint sub words of a word (Fig.2) can lead to unuseful features, thus we use a threshold for displacement at defined direction and avoid displacement higher than this threshold.

3.2 Clustering of Sub Words

Handwriting, created from sub words that sequence of them, create whole word. We consider for each of sub word, a feature vector which, will be extracted at a defined temporal window. We can define each displacement at quantized direction at time, as an impulse. We convert asynchronous but continuous flow of impulses from preprocessing unit to spatio temporal vectors. These vectors are made using spatio temporal coding and accumulation of impulses at temporal window in preprocessing phase.

Thus each vector saves impulse information (sub word information) at temporal window. Therefore at each period of time, equal to TW, we have a vector, considered as an input to clustering unit.

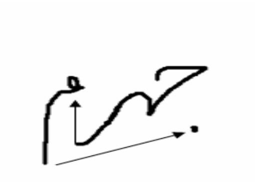


Fig. 2. Sudden jumping from one disjoint sub word to another disjoint sub word. Note that the, arrow line shows sudden jumping.

Temporal window is defined according to experimental results, obtained from dataset. Note that TW is selected so that performance of system does not decrease. Created vectors of this section are representative of a sub word (Fig.3). Sequence of these sub words vector, for each person at defined TW are behavioral identifier for a person. Now we configure SPC algorithm for this application. We get feature vector of preprocessing unit which has 8 dimensions and run SPC for different ranges of temperature T, until finding maxima of T, in which each cluster have at minimum 20 points. To run this algorithm automatically, we use a criterion based on size of clusters. First we define minimum number of data samples which must be in a cluster. This is because that increasing temperature can lead to cluster with a few point in it. In fact at high temperature the number of clusters increases [9], so we can overcome this problem, with size criterion of clusters so that we can define size of a cluster to be fraction of data set. If there isn't enough points in each cluster, we use minimum temperature. With this work we guarantee automation of method (one of advantage over Kmeans clustering) and find optimized temperature. We set parameters of SPC as below according to [9]: number of state q=20, number of nearest neighbor k=11, and

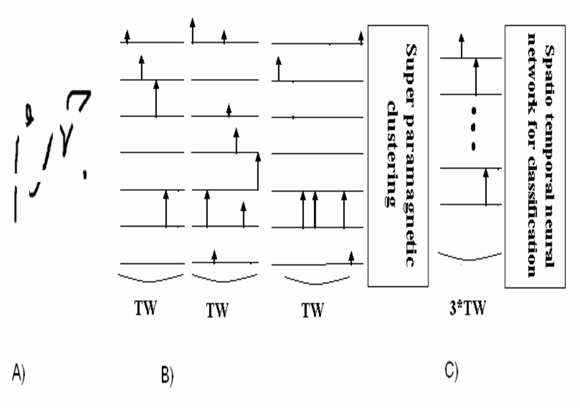

Super paramagnetic clustering
Spatio temporal neural network for classification
TW
TW
TW
3*TW
A)
B)
C)

Fig. 3. A) A word includes sequence of sub words. B) Sequence of sub words in form of impulse at determined TW. C) Feature vector of whole word for classification.

also number of iteration N=500 and correlation threshold θ=0.5. Note that changing above parameters does not affect result and classification rate depends on temperature only. Due to each iteration we change temperature T from 0 to 0.2 by step 0.01 and find maximum temperature in which each cluster included at minimum, 20 points or fraction of dataset that is determined.

3.3 Handwriting Feature Vector Classification

Feature vectors of previous section are saved according to the time of occurrence (end time of TW). In this section we can extract feature vectors of classification unit. Because of each handwriting word is a sequence of sub words at specific times, we use spatio temporal coding like preprocessing unit. Then using ST-RCE [10], [11] algorithm which is a spatio temporal neural network, classify feature vectors.

Output vector of clustering units has a dimension equal to number of clusters, that relevant element of this vector with cluster of vector is activated, when a vector of previous section come into clustering unit. Time of this event is the same as the reference time that was saved with vector. Accumulation of signals on a dimension are computed based on a reference time (here, mean sampling time of each word at database) according to (5),(6),(7). Each vector is corresponds to with a handwriting word of a person and each person corresponds to a class. Thus we can classify them using ST_RCE algorithm. This algorithm uses hermitian distance given in (8) or (9) to determine, belonging a vector to a class or not.

$$V(X,W)=\sum_{j=1}^{n}\overline{w_j}x_j \tag{8}$$

$$D(x,w)=\sqrt{\sum_{j=1}^{n}\left(x_j-w_j\right)\overline{\left(x_j-w_j\right)}} \tag{9}$$

Where D and V are distances between vector x and weight of neural network.

4 Experiments and Conclusion

In one experiment we used two databases according to table1. These databases were made using handwritings of 20 persons acquired by a tablet. We asked each person to writes 66 words generally selected from names of cities. We need enough number of samples. Thus we asked each person to write each word 7 times. At the second database for simplicity we use the first database with a little change: we force three persons to try to forge 20 words of handwriting next to original handwriting and next to original person.

We divided database to 5 dataset to train and test. Several algorithms can be used for spike clustering and best of them is an algorithm that has more discriminant strength. In order to show these differences with the proposed method we compare results of SPC clustering with Kmeans [11]. Experimental results (Table.1) shows, in cases that discrimination between spike shapes are relatively easy, two algorithms are the same, but in cases that shapes of spikes are very similar (trying to forge), SPC has acceptable advantage over Kmeans. Moreover, with SPC, we have an fully unsupervised clustering unlike Kmeans that we define number of clusters.

In another experiments we used five databases. We divide our dataset to four databases (Table.2) according to the sudden jumping in each word (Fig.2). With removing unnecessary displacement at sub words by preprocessing (jumping from disjoint sub words to another disjoint sub words of a word) the performance of classifier can increase according to (Table.2).

We found that other features beside to displacement can not increase precision of classification and moreover increase time of classification.

Note that some of the human being behaviors, like, difficulty in writing on special surface or writing with special pen can lead to undesired results, which we encountered them at the data acquisition phase. Therefore the accuracy of data acquisition, can lead to a better classification results. Unfortunately there isn't any standard database at this context and make difficult comprise of result, achieved from various method.

Table 1. Precision of methods on each type of database according to the number of forgeries

	All samples	Forgeries	With Kmeans[11]	With preprocessing	Proposed method
Database1	9240	0	71.3%	75.2%	76.2%
Database2	9240	420	68.2%	73.1%	79.1%

Table 2. Precision of methods on each databases according to the number of jumping at a word

	number of sudden jumping	Samples	With [11]	With SPC	[11]with preprocessing	Proposed method
Db1	1	18*20*7	78.2%	82.1%	82.6%	84.2%
Db2	2	17*20*7	77.2%	79.1%	80.6%	81.8%
Db3	3	12*20*7	72%	76.6%	79.4%	80.1%
Db4	4	12*20*7	69.1%	71%	73.3%	77.3%

References

1. Gupta, S.: Automatic Person Identification and Verification using Online Handwriting, thesis,Hyderabad, INDIA (2008)
2. Faundez-Zanuy, M.: On-line signature recognition based on VQ-DTW. Pattern Recognition 40, 981–992 (2007)
3. Kashi, R., Hu, J., Nelson, W.L.: A Hidden Markov Model approach to online handwritten signature verification. IJDAR 1, 102–109 (1998)
4. Thumwarin, P., Tangtisanon, P., Murata, S., Matsuura, T.: On-line Writer Recognition for Thai Numeral. In: Proc. IEEE Circuits and Systems, Asia-pacific Conference, pp. 503–508 (2002)
5. Thumwarin, P., Matsuura, T.: On-line Writer Recognition for Thai Based on Velocity of Barycenter of Pen-point Movement. In: ICIP, pp. 889–892 (2004)
6. Hangai, S., Yamanaka, S., Hamamoto, T.: On-line Signature Verification Based On Altitude and Direction of Pen Movement. In: ICME, pp. 489–492 (2000)
7. Blatt, M., Wiseman, S., Domany, E.: Super Paramagnetic Clustering of Data. Phys. Rev. Lett. 76, 3251–3254 (1996)
8. Wolf, U.: Comparison Between Cluster Montecarlo algorithm in the Ising spin model. Phys.Lett.B 228, 379–382 (1989)
9. Quiroga, Q., Nadasty, R., Ben-Shaul, Z.: Unsupervised Spike Detection and Sorting With Wavelets and Superparamagnetic Clustering. Neural Computation 16, 1661–1687 (2004)
10. Baig, R.: Spatial-Temporal Artificial Neurons Applied to On-line Cursive Handwritten Character Recognition. In: ESANN, pp. 561–566 (2004)
11. Baig, R., Hussain, M.: On-line Signture Recognition and Writer Identification Using Spatial-Temporal Processing. In: INMIC, pp. 381–385 (2004)
12. Li, B., Zhang, D., Wang, K.: Online signature verification based on null component analysis and principal component analysis. Pattern anal. applic., 345–356 (2005)
13. Zhang, K., Nyssen, E., Sahli, H.: A Multi-Stage Online Signature Verification System. Pattern Analysis & Application (5), 288–295 (2002)

Recognition and Quantification of Area Damaged by Oligonychus Perseae in Avocado Leaves

Gloria Díaz[1,⋆], Eduardo Romero[1], Juan R. Boyero[2], and Norberto Malpica[3]

[1] Universidad Nacional de Colombia, Colombia
[2] Centro de Investigación y Formación Agraria Cortijo de la Cruz, Spain
[3] Universidad Rey Juan Carlos, Spain

Abstract. The measure of leaf damage is a basic tool in plant epidemiology research. Measuring the area of a great number of leaves is subjective and time consuming. We investigate the use of machine learning approaches for the objective segmentation and quantification of leaf area damaged by mites in avocado leaves. After extraction of the leaf veins, pixels are labeled with a look-up table generated using a Support Vector Machine with a polynomial kernel of degree 3, on the chrominance components of YCrCb color space. Spatial information is included in the segmentation process by rating the degree of membership to a certain class and the homogeneity of the classified region. Results are presented on real images with different degrees of damage.

Keywords: Leaf damage, segmentation, quantification, machine learning.

1 Introduction

The persea mite, Oligonychus perseae, is a serious pest of avocado harvesting and every year it results in high economical losses for the productive sector. The presence of the mite in avocado crops is easily recognizable by the damage produced, as nearly circular necrotic regions of brownish color in the underside of leaves, distributed mainly along the central vein and other main veins (view [1]). Manual delineation of leaves can be tedious and time consuming, specially when a high number of leaves needs to be analyzed. In order to study the susceptibility of various avocado crops to the O. perseae in the south region of Spain, we wanted to quantify the mite feeding damage, using an image analysis technique to count brown spots on leaves and calculate the percentage of damaged leaf area. Several

[⋆] Gloria Díaz is supported by a grant from the Colombian Institute for the Advancement of Science and Technology (COLCIENCIAS), Grant no. 109-2005.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 677–684, 2009.

methods have already been proposed for this task. Kerguelen et al.[1] compared three methods for spot detection and obtained the best results with a simple color thresholding and heuristic constant thresholding. This procedure is not robust when different leaves have to be analyzed, or when spots of different age are considered, as it was the case for the present study. Wijekoon et al. [2] evaluate a method for quantification of fungal disease using Scion software. The method is interactive, and several parameters must be adjusted for each leave. Bakr [3] describes a software for leaf damage quantification, based on color classification of image regions. We have tested the software on our own leaves and the automatic segmentation gives very poor results (not reported here).

In the present paper, a semi-automatic approach for determining the damaged area caused by the Oligonychus perseae mite in avocado leaves is designed, implemented and validated. This is achieved in three main steps: First avocado leaves are digitized and segmented from the background using a simple Otsu thresholding [4]. A preprocessing step is applied in order to standardize the leaf color distributions and to enhance the image contrast. Then, images are segmented using a two stage color pixel classification approach: the principal veins are first extracted and the remaining pixels are then classified into damaged area or healthy leaf. Finally, the resulting segmentation is improved using a function which combines the degree of membership to the labeled class and the homogeneity of the neighborhood region.

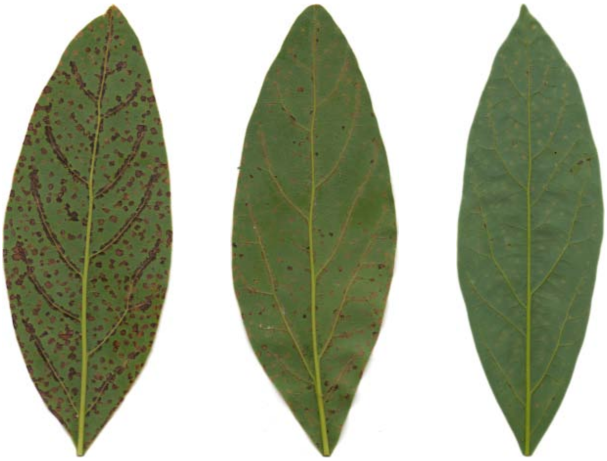


Fig. 1. Different damage levels of avocado leaves caused by Oligonychus perseae, from high (left) to low (right) damage levels. Notice how the intermediate and low levels present a very blurry regions in which, it is very difficult, even for a human observer, to establish the presence of the disease.

2 Materials and Methods

2.1 Image Acquisition

Thirteen leaves of avocado (Persea americana), of the Hass variety, damaged by Oligonychus Perseae, were harvested from a plantation in the province of Malaga, in Spain. Leaves were randomly chosen from different adult trees and scanned using and Epson Stylus DX 8400 scanner, with a resolution of 400 dpi and saved in bitmap format. Necrotic areas were manually delineated using Photoshop (Adobe Systems, San Jose, USA).

2.2 Image Preprocessing

The inherent variability of many factors, such as biological variability and acquisition conditions, results in different image luminance and color distribution in digitized leaves, which must be reduced for improving the pixel classification performance. Two major problems are addressed at this stage: contrast enhancement and color normalization between leaves. Contrast was enhanced applying a classical histogram equalization approach, whereas a color normalization approach was based on the grey world normalization assumption [5,6], which assumes that changes in the illuminating spectrum may be modeled by three constant multiplicative factors applied to the red, green, and blue color components. So, an image of unknown illumination I^u can be simply transformed to a known luminance space I^k by multiplying pixel values with a diagonal matrix $I^k_{rgb} = MI^u_{rgb}$, whit M defined as 1.

$$M = \begin{pmatrix} m_{11} & & \\ & m_{22} & \\ & & m_{33} \end{pmatrix} m_{11} = \frac{\mu^{I^k_r}}{\mu^{I^u_r}}, m_{22} = \frac{\mu^{I^k_g}}{\mu^{I^u_g}}, m_{33} = \frac{\mu^{I^k_b}}{\mu^{I^u_b}} \tag{1}$$

In this work, the color distributions of all leaves were normalized to the color distribution of one specific leaf, which was used for selecting the training points in the classification step. Figure 2, shows one example of the normalization step applied on a leaf section.

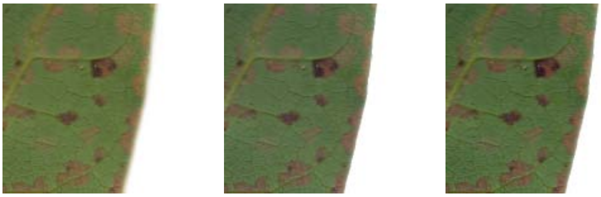


Fig. 2. Pre-processed image results. From left to right: original image, color distribution normalized and contrast enhanced.

2.3 Damaged Area Segmentation

Segmentation of the damaged area is carried out using an efficient pixel classification strategy. Given an input image I, the classification of a target pixel $I(x, y)$ is performed according to a color feature vector X describing it. In order to reduce the time needed to perform a single pixel classification, the approach presented herein is based on a classification process that find the set of boundaries that optimally separates a particular color-space representation into target classes, labeling each color component as belonging to any class. The labeled color space is used as a look-up-table (LUT) for deciding the class of each image pixel. As the selection of a color space for pixel classification is application dependent [7], we evaluated the classification performance of several color spaces and classification algorithms in order to select the most suitable for our pixel-based classification task. Four supervised classification techniques (KNN, Naive Bayes, Support Vector Machine and Multi Layer Perceptron neural networks), and four color spaces (RGB, normalized RGB, HSV and YCrCb) were assessed.

Classification models were created using a collection of the labeled pixels as training data set, which were manually extracted by an expert from a representative leaf with different damage levels (the same which was used as reference in the color normalization step). The training dataset was filtered for obtaining unique instances. Finally, less than 1000 pixels by label were used for training each classification model.

The main problem of the proposed strategy is that some colors corresponding to different damage levels are strongly mixed up with colors corresponding to leaf veins. Initially, we tried to create a learning model able to split the color spaces in two classes: damaged area and healthy leaf, but its performance was poor. We also trained some multi-class classification models, which tried to distinguish between damaged areas, veins and healthy leaf, but its performance was also poor. So, a first supervised learning model was generated for detecting the principal veins and a second one was used for classifying remaining pixels as damaged area or leaf background. Finally, a correction factor that takes the spatial relationship of the pixels into account was applied.

From the evaluated classification schemes, a SVM technique with a 3 degree polynomial kernel and the CbCr color transformation presented the best performance in the two classification stages. SVM are learning systems that project the input vectors into a high dimensional feature space, using the hypothesis space of linear functions, induced by a kernel function chosen a priori. In the feature space, the learning algorithm produces an optimal separating hyperplane between two classes, maximizing a convex quadratic form, subjected to linear constraints. The optimal hyperplane found with SVM corresponds to the maximal distance to the closest training data and is represented as a linear combination of training points called support vectors. In this work, a version of SVM that uses a sequential minimal optimization algorithm was used [8]. The classification model produces a decision class according to the distance from the instance to the optimal hyperplane. On the other hand, the $YCrCb$ is a family of color spaces commonly used to represent digital video. Luminance information is stored as a single component (Y), and chrominance corresponds to the two color-difference components (Cb and Cr). We have used the YCbCr transformation specified in the ITU-R BT.601 standard for computer-display oriented applications, but the luminance component was not considered.

Segmentation of main veins. The first stage in the segmentation approach is to extract the main leaf veins. Approaches for leaf vein extraction have been proposed previously [9,10,11,12,13]. These approaches are mainly based on the assumption that both veins and remaining leaf tend to be uniform in color and contrast [14]. Soille applied morphological filters to extract leaf veins [9]. Fu and Chi [10] proposed a two stage vein extraction that performed a preliminar segmentation based on the intensity histogram of the leaf image, which was then enhanced by a pixel classification based on edge, contrast and texture properties of the pixels. Similarly, a rough segmentation, based on the intensity histogram was used for obtaining the veins of leaf by Li et al. [15], and an active contour technique based on cellular neural network, was used to extract the veins in the obtained rough regions of leaf pixels. Li et al. [11] applied independent component analysis (ICA) to learn a set of linear basis functions that were used as patterns for vein extraction in patches of grey level leaf images. These approaches fail in our problem because damaged areas are commonly located along the veins and their color and contrast are strongly mixed. However, the color pixel classification strategy proposed was able to correctly detect the majority of main veins of the evaluated leaves.

Figure 3 displays the veins obtained when the images of Figure 1 were segmented using the LUT corresponding to veins and leaf partition of the *CrCb* color space.

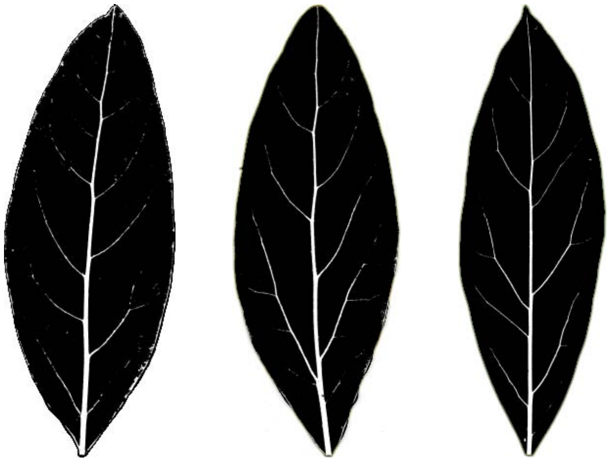


Fig. 3. Vein extraction results of leaves from Figure 1

Segmentation of damaged area. After eliminating the veins from the original images, remaining pixels were classified using the LUT generated for partitioning the color space between damaged area and healthy leaf. Figure 4 displays results of the segmentation process applied on the digitized leaves shown in Figure 1. Red color corresponds to the damaged area contour marked by the expert and white regions correspond to damaged areas found by the approach.

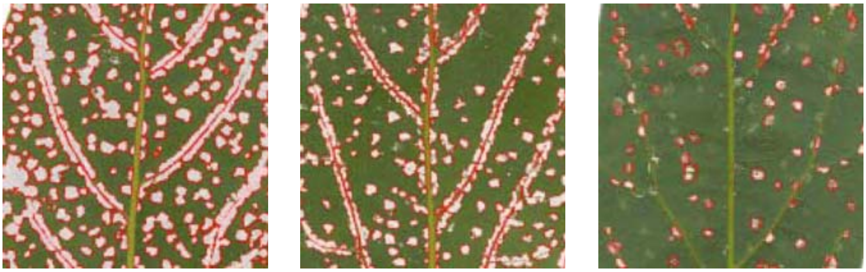


Fig. 4. Segmentation detail of damaged area of the leaves in Figure 1

Segmentation refinement. The aim of this stage was to reclassify pixels incorrectly classified, based on the use of contextual information, not taken into account by the previous phases. Pixels are reclassified using the function defined by equation 2.

$$f(p) = w_{LUT}\, l(x) + w_N\, \overline{l_{N(p)}} \tag{2}$$

where $l(x)$ stands for the label assigned to the color component x by the LUT and $\overline{l_{N(p)}}$ corresponds to the mean label value computed from a neighborhood N around pixel p. w_{LUT} and w_N are weighting functions for the label assigned by the LUT to the actual pixel and for the labels of a neighborhood N around the pixel. w_{LUT} represents a value of membership degree to the corresponding label, given by the normalized distance of the color component to the SVM hyperplane, whereas w_N represents a value of color homogeneity in a neighborhood N that is given by the difference of the color of the actual pixel with respect to N. After applying this postprocessing, some small artifacts can still be present, which are removed using a morphological opening.

2.4 Experimental Results

The proposed approach was applied to estimate the percentage of damaged area in 13 avocado leaves with different levels of damage. Damaged areas were marked by a botanist, expert in the analysis of avocado leaves. Segmentation performance was evaluated by computing average accuracy, precision and recall of damaged area segmented. Results are shown in table 1. As the measure commonly used by the botanists is the area of damaged leaf, the mean difference between the percentage of damaged area computed from the manual and automatic segmentation images is also reported in the last row of the table. The

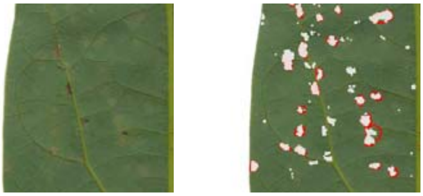


Fig. 5. Segmentation of damaged area. Detail of segmentation errors caused by poor contrast between damaged area and healthy leaf. Note that similar regions are assigned to different labels.

Table 1. Average performance of proposed approach

	Pixel-based classification	Improved Segmentation
Accuracy	0.924 ± 0.022	0.926 ± 0.022
Precision	0.703 ± 0.157	0.679 ± 0.131
Recall	0.638 ± 0.189	0.799 ± 0.171
Estimation difference	1.2 ± 2.5	2.7 ± 2.8

results show that our approach is able to detects different levels of damage in avocado leaves, produced by the Oligonychus perseae mite. The postprocessing approach improves the sensitivity of the method, although it reduces the general accuracy. Inaccuracies arose mainly due to early damage levels that are difficult to estimate even visually as shown in Figure 5. It is worth noting that differences in computed leaf damage are negligible when assigning a discrete damage level.

3 Conclusions

A simple semi-automatic approach for segmenting and quantifying damaged area in avocado leaves was proposed. The approach is based on an initial pixel based classification according to the chrominance feature from the $YCrCb$ color space, which is improved by taking into account local context information and the membership degree of the color value to a specific class (healthy leaf or damaged area). Classification time is reduced through the construction of a lookup table (LUT) in which the classes for the whole $YCrCb$ chrominance space are assigned by a learning model in an off-line process. So, a minimal user intervention is needed, to select a sample of pixels for training the color space classifier. The proposed approach was tested on 13 leaves with different color distributions and several degrees of damage and results are very promising. However, an evaluation with a higher number of leaves is warranted.

References

1. Kerguelen, V., Hoddle, M.S.: Measuring mite feeding damage on avocado leaves with automated image analysis software. Florida Entomologist 82, 119–122 (1999)
2. Wijekoon, C., Goodwin, P., Hsiang, T.: Quantifying fungal infection of plant leaves by digital image analysis using scion image software. Journal of Microbiological Methods 74, 94–101 (2008)
3. Bakr, E.M.: A new software for measuring leaf area, and area damaged by tetranychus urticae koch. Journal of Applied Entomology 129, 173–175 (2005)
4. Otsu, N.: A tlreshold selection method from gray-level histograms. IEEE Transactions on Systems, Man And Cybernetics 9, 62–66 (1979)
5. Finlayson, G.D., Schiele, B., Crowley, J.L.: Comprehensive colour image normalization. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 475. Springer, Heidelberg (1998)
6. Tek, F., Dempster, A., Kale, I.: A colour normalization method for giemsa-stained blood cell images. In: IEEE 14th Signal Processing and Communications Applications (2006)
7. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1, 67–82 (1997)
8. Platt, J.: Machines using sequential minimal optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1998)
9. Soille, P.: Morphological image analysis applied to crop field mapping. Image and Vision computing, 1025–1032 (2000)
10. Fu, H., Chi, Z.: A two-stage approach for leaf vein extraction. In: IEEE International conference on neural networks and signal processing (2003)
11. Li, Y., Chi, Z., Feng, D.D.: Leaf vein extraction using independent component analysis. In: IEEE Conference on Systems, Man, and Cybernetics (2006)
12. Nam, Y., Hwang, E., Kim, D.: A similarity-based leaf image retrieval scheme: Joining shape and venation features. Computer Vision and Image Understanding 110, 245–259 (2008)
13. Boese, B.L., Clinton, P.J., Dennis, D., Golden, R.C., Kim, B.: Digital image analysis of zostera marina leaf injury. Aquatic Botany 88, 87–90 (2008)
14. Clarke, J., Barman, S., Remagnino, P., Bailey, K., Kirkup, D., Mayo, S., Wilkin, P.: Venation pattern analysis of leaf images. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A., Meenakshisundaram, G., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T. (eds.) ISVC 2006. LNCS, vol. 4292, pp. 427–436. Springer, Heidelberg (2006)
15. Li, Y., Zhu, Q., Cao, Y., Wang, C.: A leaf vein extraction method based on snakes technique. In: International Conference on Neural Networks and Brain (2005)

Neurocontroller for Power Electronics-Based Devices

M. Oliver Perez[1], Juan M. Ramirez[1], and H. Pavel Zuniga [2]

[1] CINVESTAV- Guadalajara. Av. Científica 1145. Zapopan, Jalisco, 45015. Mexico
{operez,jramirez}@gdl.cinvestav.mx
[2] Universidad de Guadalajara. Centro Universitario de Ciencias Exactas e Ingeniería.
Posgrado de Ingeniería Eléctrica. Guadalajara, Jal., Mexico
pavel.zuniga@cucei.udg.mx

Abstract. This paper presents the Static Synchronous Compensator's (StatCom) voltage regulation by a B-Spline neural network. The fact that the electric grid is a non-stationary system, with varying parameters and configurations, adaptive control schemes may be advisable. Thereby the control technique must guarantee its performance on the actual operating environment where the StatCom is embedded. An artificial neural network (ANN) is trained to foresee the device's behavior and to tune the corresponding controllers. Proportional-Integral (PI) and B-Spline controllers are assembled for the StatCom's control, where the tuning of parameters is based on the neural network model. Results of the lab prototype are exhibited under different conditions.

Keywords: Artificial neural network, B-Spline, StatCom, FACTS.

1 Introduction

Power systems are highly nonlinear, with time varying configurations and parameters [1-3]. Thus, PI controllers based on power system's linearized model cannot guarantee a satisfactory performance under broad operating conditions. Thus, in this paper the use of a control, adjustable under different circumstances, is suggested.

StatCom requires an adaptive control law which takes into account the nonlinear nature of the plant and adapts to variations of the environment to regulate the bus voltage magnitude. The aim of this paper is the utilization of an adaptive B-Spline neural network controller. The fundamental structure of such device is based on a Voltage Source Converter (VSC) and a coupling transformer, which it is used as a link with the electric power system, Fig. 1. E_{ST} represents the StatCom's complex bus voltage, and E_k the power system complex bus voltage [4-7]; all angles are measured with respect to the general reference.

The model is represented as a voltage variable source E_{ST}. Its magnitude and phase angle can be adjusted with the purpose of regulating the bus voltage magnitude. The magnitude V_{ST} is conditioned by a maximum and a minimum limit, depending on the VSC's capacitor rating.

In this paper a B-Spline neurocontroller is utilized due to its ability to adapt its performance to different operating conditions. A PI controller is also utilized for comparison purposes. Tuning the prototype's controllers is a tedious task since

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 685–692, 2009.

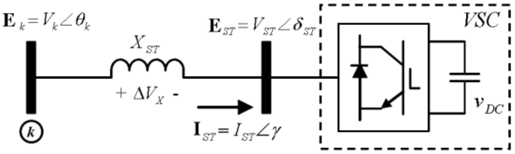

$\mathbf{E}_k = V_k\angle\theta_k$
X_{ST}
$+ \Delta V_X$ -
k
$\mathbf{E}_{ST} = V_{ST}\angle\delta_{ST}$
$\mathbf{I}_{ST} = I_{ST}\angle\gamma$
VSC
v_{DC}

Fig. 1. StatCom's schematic representation

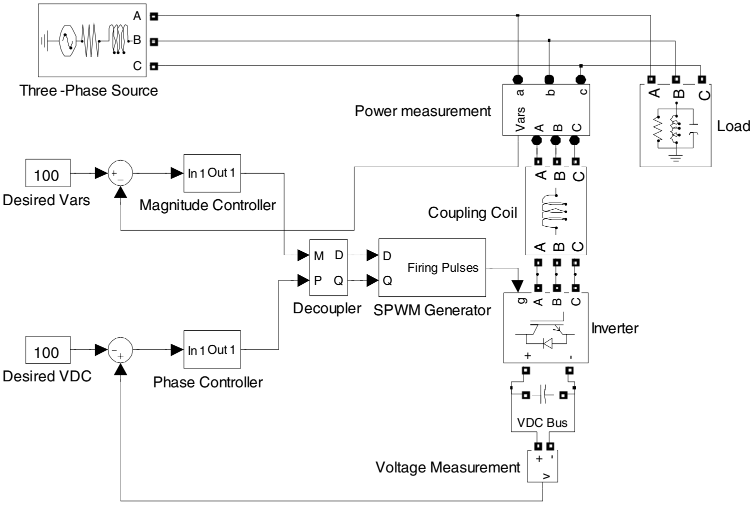


Fig. 2. Assembled prototype block diagram

trial-and-error strategy may be a long process. Thus, a StatCom´s model is developed to haste all tests and predict the device's behavior.

B-Spline controller is chosen because the PI is not self adaptable when the operating condition change. If the operating condition is changed the PI will not function properly because it would be out of the region for which it is designed. Once the B-Spline controller is designed and its effectiveness is tested by simulation [8]-[9], it is assembled.

A StatCom-based SPWM is a multiple-input multiple-output (MIMO) system. Its input signals are the magnitude and phase of the 60-Hz modulating signal in conjunction with a 3000-Hz triangle carrier signal generate the six firing signals to operate every gate of a six-IGBT inverter. Two output signals are controlled: (*a*) the reactive power flowing into or out of the device and, (*b*) the capacitor's DC voltage, Fig. 2. Thus, in this paper the reactive power flow is controlled through the amplitude of the modulating signal, while the DC voltage is controlled through the phase of the modulating signal.

2 B-Spline Neural Networks: A Summary

The major advantages of ANN-based controllers are simplicity of design, and their compromise between complexity and performance. The B-SNN is a particular case of neural networks that are able to adaptively control a system, with the option of carrying out such tasks on-line, taking into account non-linearities [10-12].

Additionally, through B-SNN it is possible to bound the input space by the basis functions' definition. The most important feature of the B-Spline algorithm is the output's smoothness that is due to the shape of the basis functions. The bus voltage magnitude must attain its reference value through the B-Spline adaptive control scheme. That is, control must drive the StatCom's modulation ratio *m* and the phase angle α to the desired value in order to regulate the injected voltage of the shunt converter.

The B-Spline neural network output can be expressed as [13],

$$y = \sum_{i=1}^{p} a_i w_i \tag{1}$$

where w_i and a_i are the i^{th} weight and the i^{th} B-spline basis function output, respectively; *p* is the number of weights. Let us define:

$$\mathbf{w} = [w_1 \; w_2 \ldots w_p]^T, \qquad \mathbf{a} = [a_1 \; a_2 \ldots a_p]^T$$

Thereby, eqn. (1) can be rewritten as:

$$\mathrm{y} = \sum_{i=1}^{p} a_i w_i = a^T w \tag{2}$$

The last expression can be rewritten in terms of time as:

$$\mathrm{y}(t) = a^T(t)w(t-1) = a^T\big(x(t)\big)w(t-1) \tag{3}$$

where *a* is a p-dimensional vector which contains the function basis outputs, *w* is the vector containing the weights, and *x* is the input vector.

Learning in artificial neural networks (ANNs) is usually achieved by minimizing the network's error, which is a measure of its performance, and is defined as the difference between the actual output vector of the network and the desired one.

On-line learning of continuous functions, mostly via gradient based methods on a differentiable error measure is one of the most powerful and commonly used approaches to train large layered networks in general [13], and for non stationary tasks in particular. In this paper, the neurocontroller is trained on-line using the following error correction instantaneous learning rule [14],

$$\Delta w(t-1) = \frac{\gamma e_y(t)}{\|a(t)\|_2^2} a(t) \tag{4}$$

where: γ is the learning rate and $e_y(t)$ is the instantaneous output error.

The proposed neurocontroller consists fundamentally on establishing its structure (the definition of basis functions) and the value of the learning rate. Regarding the weights' updating, (4) should be applied for each input-output pair in each sample time; the updating occurs if the error is different from zero. Hence, the B-SNN training process is carried out continuously on-line, while the weights' value are updated using the feedback variables. The proposed controller is based on (4). Inside the Spline block the activation function is located; in this case an Spline function.

3 Test Results

A lab StatCom prototype has been implemented in order to validate the proposition. The major elements of the scheme are the following, Fig. 3: (*i*) source voltage – 85 volts RMS, (*ii*) transmission line inductance – 3.1 mH, (*iii*) LC filter – Capacitors 5μF and inductors 3.1 mH, (*iv*) asynchronous motor – squirrel cage 1.5 HP. The Voltage Source Converter (VSC), which is the main component, has been controlled by a DSP TMS320F2812. This DSP possesses 6.67 ns instruction cycle time (150 MHZ), 16 channel, 12-bit ADC with built-in dual sample and hold, an analog input from 0 to 3 V.

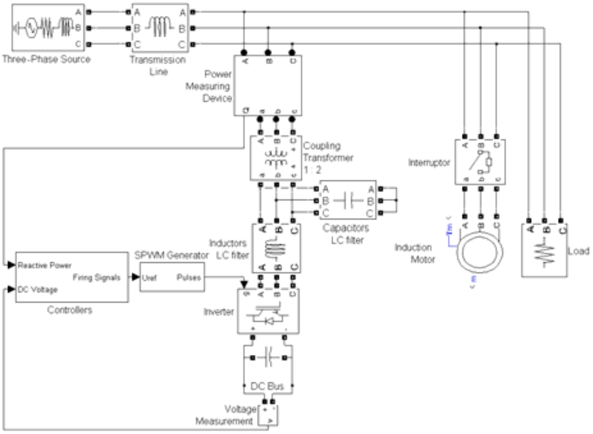


Fig. 3. Circuit arrangement

The synchronizing circuit utilized for the six IGBT VSC has been implemented in the DSP, collecting the data with a global Q of 11, which means that it reserves 21 bits for the data´s integer part and 11 bits for the fractional one. In this application the selected sampling frequency is 3000 Hz, thus 50000 clock cycles available between successive samples can be accomplished. In open loop, reactive power and DC voltage measurements are carried out. Feeding this set of measurements into the 40,60,2 scheme feed forward neural network, back propagation type, and training the created network by 800 epochs, a suitable model of the prototype is accomplished. The proposed ANN which will simulate the prototype Statcom is a 40,60,2 scheme feed forward, BP type. It means that it will have a 40 neurons first layer, a middle layer of 60 layers of a sigmoid transfer function, and two neurons in the output layer. The ANN is trained with four vectors of 229 elements each, two vectors for the input and two vectors for the output.

3.1 Proportional-Integral Controller

Firstly, two PI controllers are tried: (a) one for the reactive power flow, and (b) one for the DC voltage, Fig. 3. Two different conditions are analyzed:

(a) *Case 1*. The outputs' reference values are: 100 Vars flowing outward the StatCom, and 97.92 DC volts at the inverter´s capacitors.

(b) *Case 2*. The outputs' reference values are: 114 Vars flowing outward the StatCom, and 97.00 DC volts at the inverter´s capacitors.

In this case, the same controller structure is employed for both loops. To tune the PI's controller parameters is the first objective. Its structure is defined as follows,

$$\frac{y(s)}{u(s)} = K_p + \frac{K_i}{s} \tag{5}$$

In such process an intensive use of the ANN previously trained is done. The following parameters produce under damped response without overshoots: K_{im} = 0.9; K_{pm} = 2.0; K_{if} = 3e-4; K_{pf} = 1.0e-3. K_{im} is the integral gain and K_{ip} is the proportional gain for the magnitude controller. K_{if} and K_{pf} are the gains for the phase controller, respectively. The system is feeding the resistive load only, Fig. 3. Fig. 4 depicts the reactive power and DC voltage obtained by simulation. The physical realization is displayed in Fig. 5. At t = 19 s the induction motor is started and turned out immediately. At t = 29 s it is started again and after several cycles it is turned out. Notice that during this time output signals do not reach their reference value. Under this condition the amplitude and phase of the modulating signal reach their maximum.

However, if the desired values and the initial state are modified, the PI controlled StatCom ´s output exhibits a different behavior, Fig. 6. In this simulation the desired values are 114 Vars delivered and 97.00 DC volts. Now, the new initial state, *Case 2*, is such that the output voltage lags 1.8 degrees with respect to the grid´s voltage, by a modulation index of 90%.

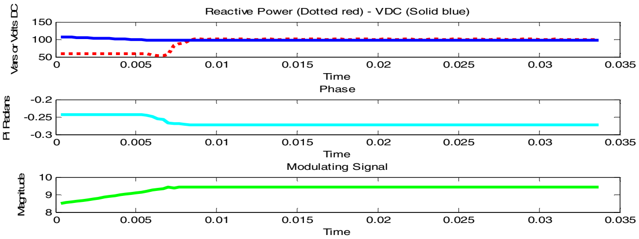


Fig. 4. *Case 1*. Simulated ANN response for Kim = 0.9; Kpm = 2.0; Kif = 3e-4; Kpf = 1.0e-3. From top to bottom: (*a*) reactive power and DC voltage, (*b*) phase, and (*c*) magnitude of the modulating signal.

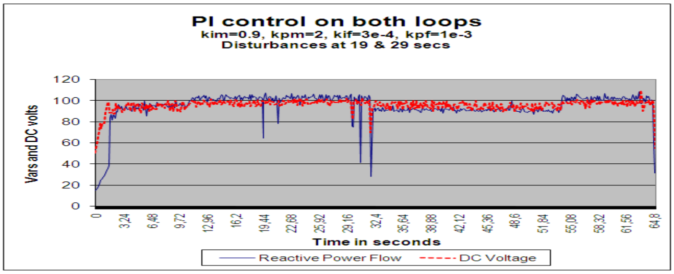


Fig. 5. *Case 1*. Prototype's response (Var and DC voltage) for Kim = 0.9; Kpm = 2.0; Kif = 3e-4; Kpf = 1.0e-3

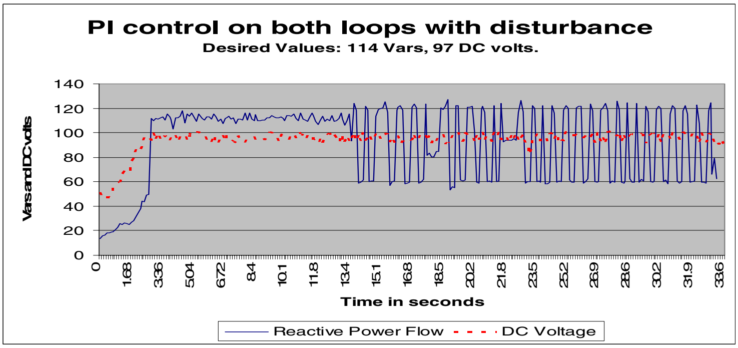


Fig. 6. *Case 2*. Prototype's response. PI behavior under a disturbance.

Then, with different initial states the PI controllers are tested. Under Case 2, a fast change in the DC voltage due to the induction motor starting compels the system to oscillate, Fig. 6. Hence, the tuned PI parameters that exhibited a satisfactory performance in Fig. 4 are not able to work well when the StatCom migrates to another operating point.

3.2 B-Spline Controller

The proposed B-Spline controller is now simulated with the StatCom´s ANN model. Originally, the desired values are as in the PI case (*initial state*): 100 Vars flowing outward the StatCom, and 97.92 DC volts at the inverter´s capacitors. The StatCom´s *initial state* generates an output voltage lagged 0.4 degrees with respect of the grid´s voltage; the inverter´s output voltage presents a modulation index of 85%. Fig. 7 shows that the references are reached with both loops based on B-Spline controllers. The slower loop is the DC voltage loop; it is handled through the learning factor Nf. In the present case Nf = 0.1. Both desired values are reached in 35 ms and the response signal exhibits an overshoot. The responses with PIs are improved, Fig. 4.

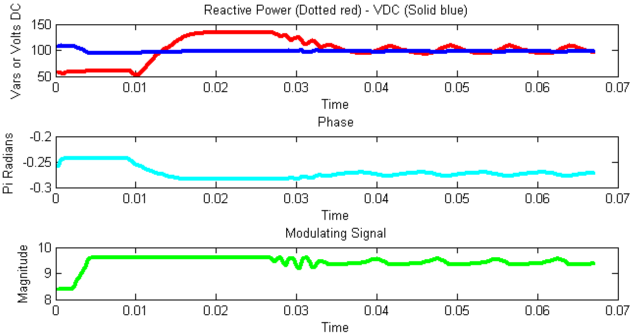


Fig. 7. *Case 1*. Simulated ANN B-Spline response for Wm=12636, Nm=40, Wf=-0.3621, Nf=0.1. From top to bottom: (*a*) reactive power and DC voltage, (*b*) phase, and (*c*) magnitude of the modulating signal.

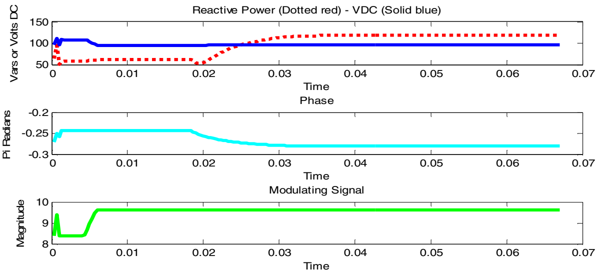


Fig. 8. *Case 2*. Simulated ANN B-Spline response for Wm=12636, Nm=40, Wf=-0.3621, Nf=0.1. From top to bottom: (*a*) reactive power and DC voltage, (*b*) phase, and (*c*) magnitude of the modulating signal.

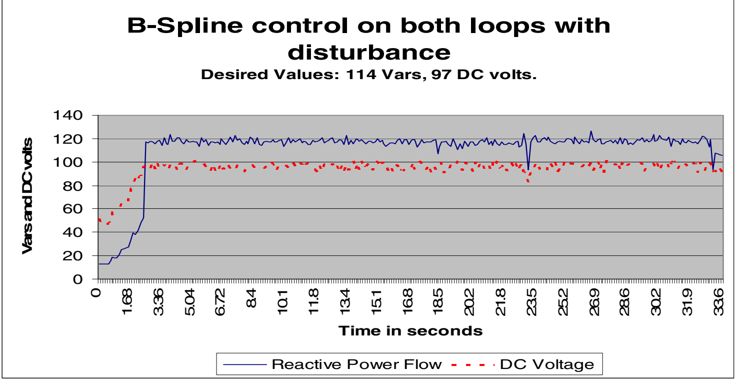


Fig. 9. *Case 2*. Prototype's B-Spline response (Vars and DC voltage) showing its adaptability

A new initial state is taken into account: 114 Vars delivered and 97.00 DC volts. Under this initial state the StatCom exhibits the output behavior in Fig. 8.

The PI's parameters for an *initial state* may not be the appropriate ones for another one. In this example the StatCom gets into a non stable region and the B-Spline controller exhibits a slower response compared with the first *initial state*; the desired values are reached without stability problems. The B-Spline performance is depicted in Fig. 9. The controllers' parameters are the same: for the magnitude controller an initial condition of 12636 DSP units and learning factor of 400 DSP units, while for the phase controller an initial condition of -0.3621 pi radians and a learning factor (Nf) of 0.1. Both references are attained. Finally, the B-Spline controller's performance displayed in Fig. 9 proved the adaptability of the B-Spline controller by rejecting the disturbance related to the starting motor on a different operational point respect to the one it is tuned. At t = 23 s and at t = 33 s, the induction motor is started.

4 Conclusions

The proposed neurocontroller represents a pertinent choice for on-line control due to it possesses learning ability and fast adaptability, robustness, simple control algorithm, and fast calculations. Unlike the PI control technique, the B-Spline NN control exhibits adaptive behavior since the weights can be adapted on-line responding to inputs and error values as they take place. These are desirable characteristics for practical hardware on power station platforms. Simulating the StatCom´s behavior with an ANN reduces the tuning time and offers a predictive view of the systems response. Lab results for different disturbances and operating conditions demonstrate the effectiveness and robustness of the NN control.

References

1. Mohagheghi, S., Park, J.-W.: Adaptive critic design based neurocontroller for a STATCOM connected to a power system. In: IEEE Industry Applications Conference, 38th IAS Annual Meeting, vol. 2, pp. 749–754 (2003)
2. Mohagheghi, S., Venayagamoorthy, G.K., Harley, R.G.: Adaptive Critic Design Based Neuro Fuzzy Controller for a Static Compensator in a Multimachine Power System. IEEE Transactions on Power Systems 21(4), 1744–1754 (2006)
3. Abido, M.A., Abdel-Magid, Y.L.: Radial basis function network based power system stabilizers for multimachine power systems. In: International Conference on Neural Networks, June 9–12, vol. 2, pp. 622–626 (1997)
4. Song, Y.H., Johns, A.T.: Flexible AC Transmission System (FACTS). The Institution of Electrical Engineers, United Kingdom (1999)
5. Acha, E., Fuerte-Esquivel, C.R., Ambriz-Pérez, H., Ángeles Camacho, C.: FACTS: Modelling and Simulation in Power Network. John Wiley & Sons, LTD, England (2004)
6. Lehn, P.W., Iravani, M.R.: Experimental Evaluation of STATCOM Closed Loop. IEEE Trans. on Power Delivery 13(4), 1378–1384 (1998)
7. Wang, H.F.: Applications of damping torque analysis to STATCOM control. In: Electrical Power and Energy Systems, vol. 22, pp. 197–204. Elseiver, Amsterdam (2000)
8. Olvera, R.T.: Assembling of the B-Spline Neurocontroller to regulate the Statcom (in Spanish). A dissertation for the degree of Doctor of Sciences (December 2006), `http://www.dispositivosfacts.com.mx/doctos/doctorado/RTO_tesis_doctorado.pdf`
9. Haro, P.Z.: Analysis and control of a series compensator. A dissertation for the degree of Doctor of Sciences (May 2006), `http://www.dispositivosfacts.com.mx/doctos/doctorado/Tesis_pavel2006.pdf`
10. Cong, S., Song, R.: An Improved B-Spline Fuzzy-Neural Network Controller. In: Proc. 2000 3rd World Congress on Intelligent Control and Automation, pp. 1713–1717 (2000)
11. Cheng, K.W.E., Wang, H.Y., Sutanto, D.: Adaptive directive neural network control for three-phase AC/DC PWM converter. In: IEE Proc. Electr. Power Appl., September 2001, vol. 148, pp. 425–430 (2001)
12. Reay, D.S.: CMAC and B-spline Neural Networks Applied to Switched Reluctance Motor Torque Estimation and Control. In: 29th Annual Conf., IEEE Industrial Electronics Society, pp. 323–328 (2003)
13. Brown, M., Harris, C.J.: Neurofuzzy Adaptive Modelling and Control. Prentice Hall International, Englewood Cliffs (1994)
14. Cedric Yiu, K.F., Wang, S., Teo, K.L., Tsoi, A.C.: Nonlinear System Modeling via Knot-Optimizing B-Spline Networks. IEEE Trans. Neural Networks 12, 1013–1022 (2001)

XII Keynote 4

3D and Appearance Modeling from Images

Peter Sturm[1], Amaël Delaunoy[1], Pau Gargallo[2], Emmanuel Prados[1],
and Kuk-Jin Yoon[3]

[1] INRIA and Laboratoire Jean Kuntzmann, Grenoble, France
[2] Barcelona Media, Barcelona, Spain
[3] GIST, Gwangju, South Korea

Abstract. This paper gives an overview of works done in our group on 3D and appearance modeling of objects, from images. The backbone of our approach is to use what we consider as the principled optimization criterion for this problem: to maximize photoconsistency between input images and images rendered from the estimated surface geometry and appearance. In initial works, we have derived a general solution for this, showing how to write the gradient for this cost function (a non-trivial undertaking). In subsequent works, we have applied this solution to various scenarios: recovery of textured or uniform Lambertian or non-Lambertian surfaces, under static or varying illumination and with static or varying viewpoint. Our approach can be applied to these different cases, which is possible since it naturally merges cues that are often considered separately: stereo information, shading, silhouettes. This merge naturally happens as a result of the cost function used: when rendering estimated geometry and appearance (given known lighting conditions), the resulting images automatically contain these cues and their comparison with the input images thus implicitly uses these cues simultaneously.

1 Overview

Image-based 3D and appearance modeling is a vast area of investigation in computer vision and related disciplines. A recent survey of multi-view stereo methods is given in [6]. In this invited paper, we provide a brief overview of a set of works done in our group, mainly by showing sample results. Technical details can be found in the relevant cited publications.

3D and appearance modeling from images, like so many estimation problems, is usually formulated, explicitly or implicitly, as a (non-linear) optimization problem[1]. One of the main questions is of course which criterion to optimize. We believe that the natural criterion is to maximize photoconsistency between input images and images rendered from the estimated surface geometry and appearance (to be precise, this criterion corresponds to the likelihood term of a Bayesian problem formulation, which can be combined with suitable priors). To measure

[1] There exist some exceptions in special cases. For example, in basic shape-from-silhouettes, the 3D shape is directly defined by the input and no estimation is necessary, just a computation to explicitly retrieve the shape.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 695–704, 2009.

photoconsistency, one may use for example the sum of squared differences of grey levels or the sum of (normalized) cross-correlation scores. This criterion is simple to define but turns out to be hard to optimize rigorously. To optimize it we process a gradient descent. When speaking about gradient descent, a central question is how to compute the gradient of the criterion. Yezzi and Soatto have shown how to do so, but only for convex objects [7]. In [3], we developed the gradient for the general case. Importantly, it correctly takes into account how surface parts become visible or invisible in input cameras, due to the surface evolution driven by the gradient. Hence, using this gradient, silhouettes and apparent contours are implicitly handled correctly since these are the places where such visibility changes take place. Further, due to comparing input with rendered images, color and shading effects are also naturally taken into account. Overall, rigorously optimizing the photoconsistency between input and rendered images, allows to naturally merge stereo, shading, and silhouette cues, within a single framework and without requiring tuning parameters to modulate their relative influence.

This framework was first developed for a continuous problem formulation [3] (we used level sets for the surface parametrization). We then developed it for the case of discrete surface representations, in particular triangular meshes [2] which in practice allow to achieve a higher 3D surface resolution. Also, even when using a continuous setup, in practice the surface representation is finally discretized and the surface evolution requires to repeatedly discretize attributes. It thus seems more natural to directly start with a discrete parametrization and do all derivations based on it. In both cases, continuous and discrete, the surface evolution can be carried out by gradient descent (one may also try less basic methods, such as conjugate gradient, quasi-Newton methods etc.).

The developed framework for optimizing photoconsistency was then used to develop a general purpose algorithm for modeling 3D surface and appearance [8,9]. Here, we considered the case where lighting conditions are known (we modeled this as a set of point or directional light sources, plus an ambient lighting) but may be different for each input image. The most general instance of our algorithm estimates an object's 3D surface and a spatially varying appearance. For the latter, we use the standard Blinn-Phong reflectance model and can in principle estimate one set of reflectance coefficients (albedo and specular coefficients) per surface point, allowing to reconstruct non-Lambertian surfaces. However, estimating specular coefficient for each point is obviously highly ill-posed, so the most general experiment we carried out used a strong smoothness prior over these coefficients.

This general algorithm can be run on more constrained examples, in principle simply by leaving out the appropriate parts in the problem parametrization and the computation of cost function, gradient, etc. Examples of some scenarios are given in the following section. For example, one may model the surface appearance by a spatially varying albedo plus uniform specular coefficients, by a spatially varying albedo and no specular effects or simply by a uniform albedo. In the case of constant lighting, the second case corresponds to multi-view stereo whereas the third case corresponds to (multi-view) shape-from-shading. Also, if variable lighting conditions are considered but a static viewpoint, the algorithm will perform photometric stereo, whereas in the general case of varying lighting and viewpoint, one finds a combination of multi-view and photometric stereo.

2 Sample Scenarios and Results

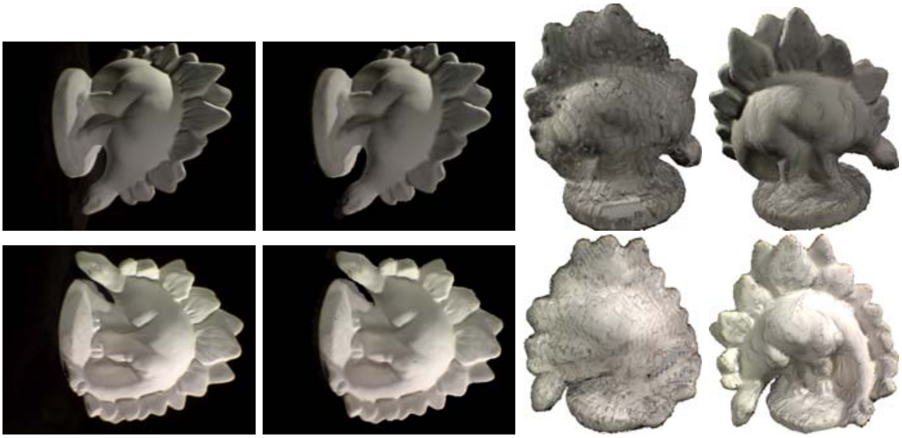
As mentioned above, due to the generality of the proposed approach, it can be applied to various types of image sets with different camera/light configurations. Here, knowledge of illumination allows to factorize radiance into reflectance and geometry. In practice, depending on the scenario, that knowledge may not be required, e.g. for recovering shape and radiance of Lambertian surfaces with static illumination. In other words, when images of Lambertian surfaces are taken under static illumination, the proposed approach can be applied even without lighting information, assuming that there is only an ambient illumination. In this case, the approach works much like the conventional multi-view stereo methods and estimates the shape and radiance of Lambertian surfaces. Figure 1 shows the result for the dino image set [6], for which no lighting information is required. The proposed method successfully recovers the shape as well as the radiance.

In the following, for synthetic data sets, the estimated shape is quantitatively evaluated in terms of accuracy and completeness as in [6]. We used 95% for accuracy and the 1.0mm error for completeness. For easy comprehension, the size of a target object is normalized so that it is smaller than [100mm 100mm 100mm]. Here, beside the shape evaluation, we also evaluated the estimated reflectance in the same manner. For each point on an estimated surface, we found the nearest point on the true surface and compute the distance and reflectance differences, and vice versa.

The proposed approach can also be applied to images taken under varying illumination. Results using images of textureless/textured Lambertian surfaces are shown in Figs. 2 to 5. Figure 2 shows the ground-truth shape of the "bimba" image set (18 images) of a textureless object, and the estimation result. The surface has uniform diffuse reflectance and input images were taken under different illuminations. In this case, the approach works as a multi-view photometric stereo method and recovers the shape and the diffuse reflectance of each surface point. Here, black points in the estimated model correspond to points that were not visible from any camera and/or any light source.

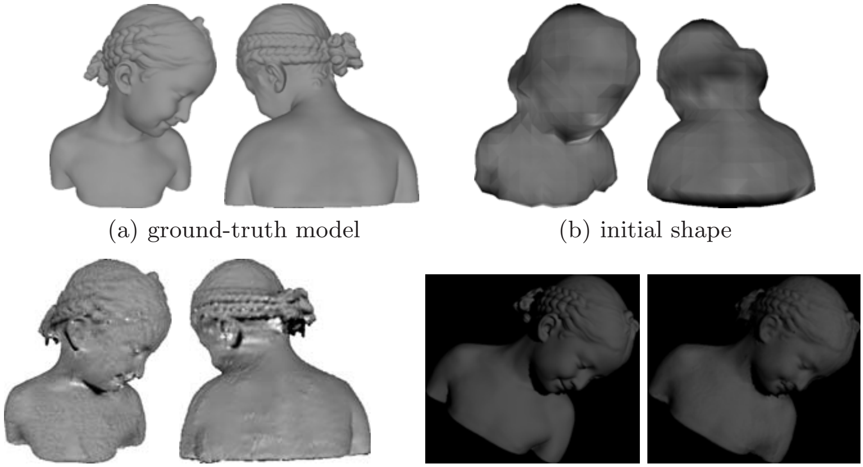
Results for a more complex object are shown in Figs. 3 and 4. The images synthesized using the estimation closely resemble input images while the shading and the reflectance are successfully separated. Furthermore, it is possible to synthesize images under different lighting conditions, even from different viewpoints. The proposed method also recovers concave parts well as shown in Fig. 5.

We also applied our approach to the images of textureless/textured *non-Lambertian* surfaces showing specular reflection. Note that, unlike previous methods [1,4], we do not use any thresholding to filter out specular highlight pixels. The result for the smoothed "bimba" data set is shown in Fig. 6. In this case, the surface



(a) input images (b) synthesized images (c) initial shape (d) estimated shape

Fig. 1. Result for the "dino" image set (16 images) — Lambertian surface case (static illumination and varying viewpoint)



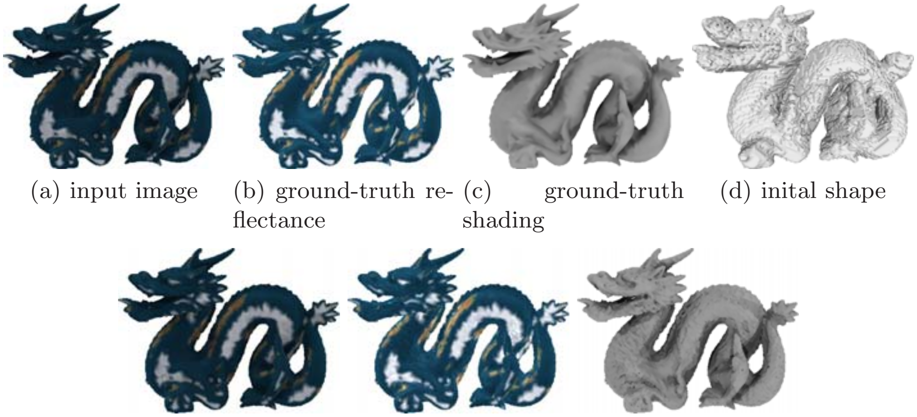
(a) ground-truth model

(b) initial shape

(c) estimated model

(d) input image vs. synthesized image

Fig. 2. Result for the "bimba" image set (18 images) — textureless Lambertian surface case (varying illumination and viewpoint). 95% accuracy (shape, ρ_{dr}, ρ_{dg}, ρ_{db})=(2.16mm, 0.093, 0.093, 0.093), 1.0mm completeness (shape, ρ_{dr}, ρ_{dg}, ρ_{db}) = (82.63%, 0.104, 0.104, 0.104).



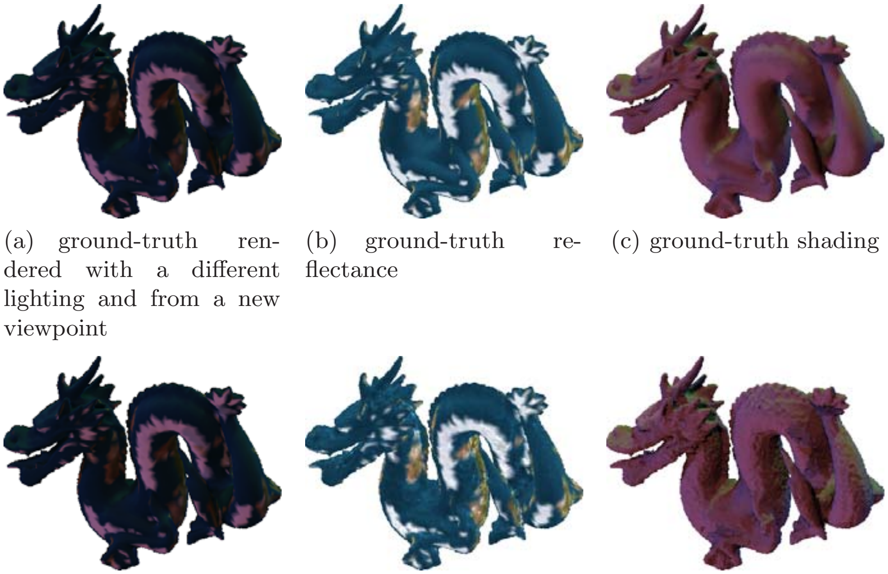
(a) input image (b) ground-truth reflectance (c) ground-truth shading (d) inital shape

(e) synthesized image (f) estimated reflectance (g) estimated shading

Fig. 3. Result for the "dragon" image set (32 images) — textured Lambertian surface case (static illumination and varying viewpoint). 95% accuracy (shape, ρ_{dr}, ρ_{dg}, ρ_{db})=(1.28mm, 0.090, 0.073, 0.066), 1.0mm completeness (shape, ρ_{dr}, ρ_{dg}, ρ_{db}) = (97.11%, 0.064, 0.056, 0.052).

has uniform diffuse/specular reflectance and each image was taken under a different illumination. Although there is high-frequency noise in the estimated shape, the proposed method estimates the specular reflectance well. Note that most previous methods do not work for image sets taken under varying illumination and, moreover, they have difficulties to deal with specular reflection even if the images are taken under static illumination. For example, Fig. 7 shows a result obtained by the method of [5] and our result for comparison. We ran the original code provided by the authors many times while changing parameters and used mutual information (MI) and cross correlation (CCL) as similarity measures to get the best results under specular reflection. As shown in Fig. 7, the method of [5] fails to get a good shape even when the shape is very simple, while our method estimates it accurately. Also, with such images, given the large proportion of over-bright surface parts, it seems intuitive that the strategy chosen by [1] and [4] (who consider bright pixels as outliers) might return less accurate results, because it removes too much information.

We also used real image sets of textured glossy objects, which were taken by using fixed cameras/light sources, while rotating the objects as in [1,4] — in this case, each image has a different illumination and observes specular reflections. The light position and color were measured using a white sphere placed in the scene. Figure 8 shows one image among 59 input images, the initial shape obtained using silhouettes, and the final result. Here, we simply assumed a single-material surface (i.e. uniform specular reflectance, but varying albedo). Although a sparse grid volume was used, the proposed method successfully estimated the shape of the glossy object even under specular reflection, while estimating the latter. Here, we can see that, although the estimated specular reflectance may



(a) ground-truth rendered with a different lighting and from a new viewpoint

(b) ground-truth reflectance

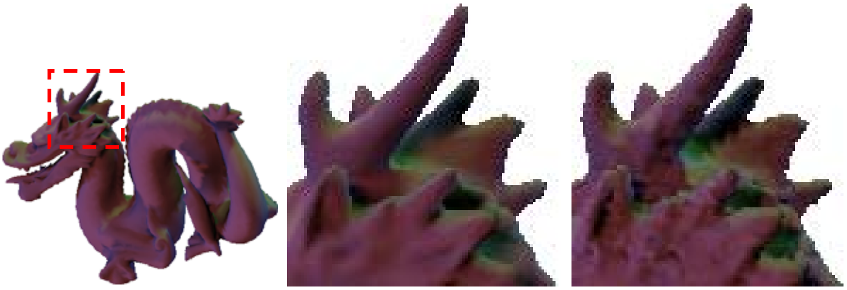
(c) ground-truth shading

(d) reconstructed scene relighted as in (a) and viewed from the same point of view as (a)

(e) estimated reflectance

(f) estimated shading

Fig. 4. Synthesized result for different lighting conditions and viewed from a viewpoint that is different from all input viewpoints. A comparison with the ground-truth is possible because this is synthetic data.

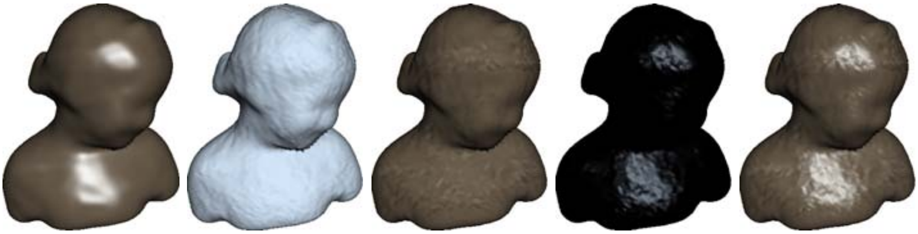


(a) ground-truth

(b) close-up view

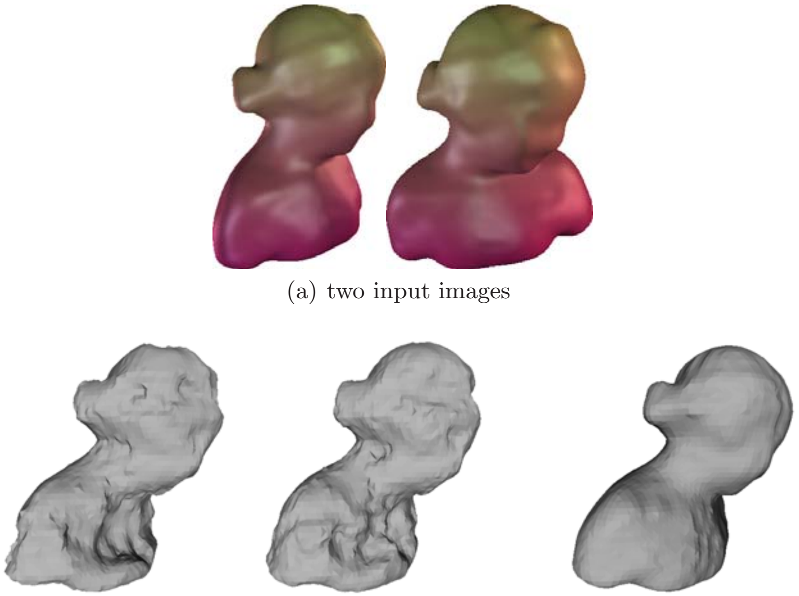
(c) close-up view of the estimated result

Fig. 5. Close-up view of the concave part of the "dragon" model



(a) ground-truth model (b) estimated shape (c) diffuse image (d) specular image (e) synthesized image

Fig. 6. Result for the smoothed "bimba" image set (36 images) — textureless non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint). 95% accuracy (shape, ρ_{dr}, ρ_{dg}, ρ_{db}, ρ_s, α_s)=(0.33mm, 0.047, 0.040, 0.032, 0.095, 8.248), 1.0mm completeness (shape, ρ_{dr}, ρ_{dg}, ρ_{db}, ρ_s, α_s) = (100%, 0.048, 0.041, 0.032, 0.095, 8.248).

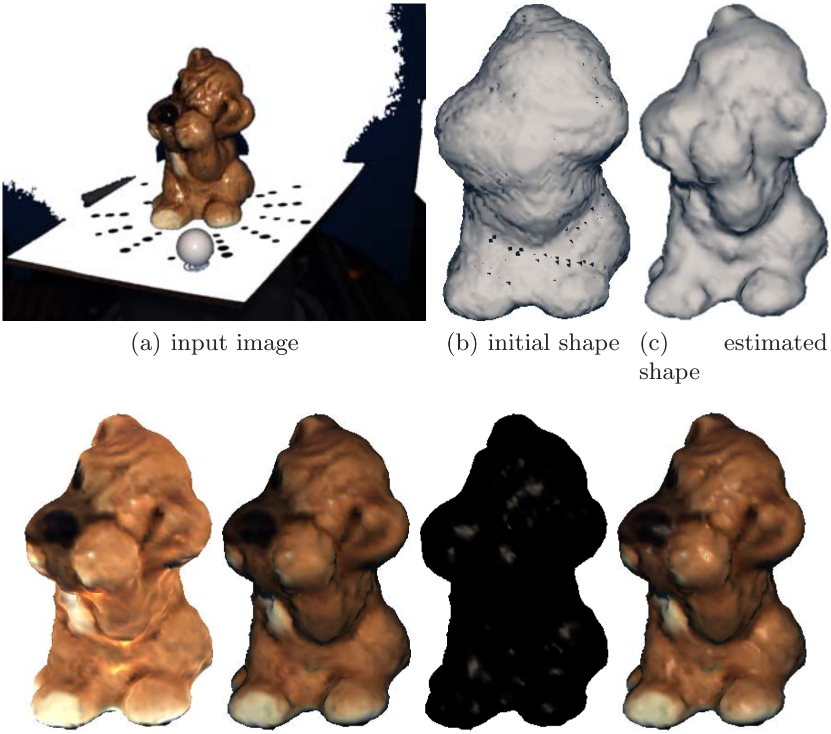


(a) two input images

(b) results using [5] (MI and CCL)

(c) our result

Fig. 7. Result comparison using the smoothed "bimba" image set (16 images) — textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint)



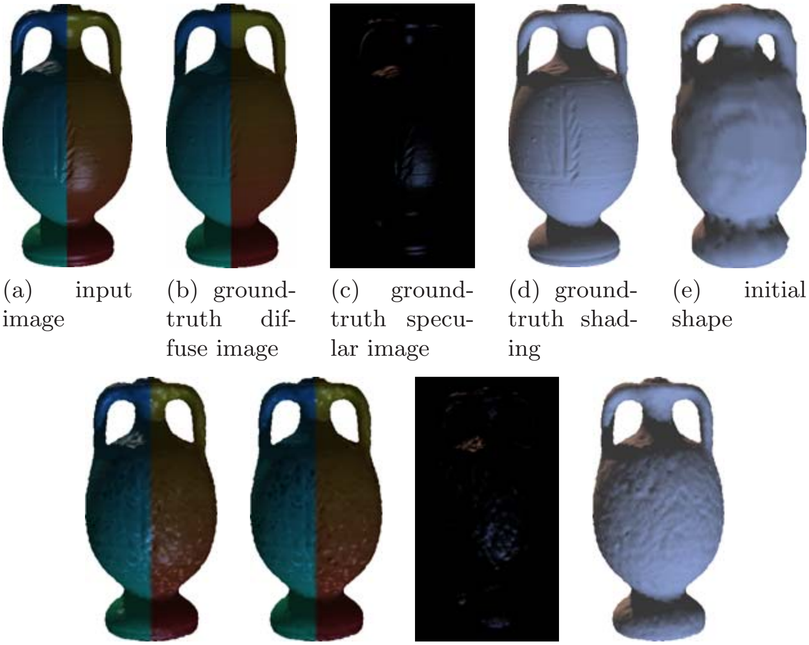
(a) input image (b) initial shape (c) estimated shape

(d) diffuse reflectance (e) diffuse image (f) specular image (g) synthesized image

Fig. 8. Result for the "saddog" image set (59 images) — textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint)

not be highly accurate because of the inaccuracy of lighting calibration, saturation, and unmodeled photometric phenomena such as interreflections that often occur on glossy surfaces, it really helps to recover the shape well.

Finally, we applied our approach to the most general case — images of textured non-Lambertian surfaces with spatially varying diffuse and specular reflectance and shininess, cf. Fig. 9. Input images were generated under static illumination (with multiple light sources) while changing the viewpoint. Figure 9 shows one image among 36 input images, one ground-truth diffuse image, one ground-truth specular image, ground-truth shading, and our results. We can see that the proposed method yields plausible specular/diffuse images and shape. However, there is high-frequency noise in the estimated shape. Moreover, the error in reflectance estimation is rather larger compared to the previous cases because of sparse specular reflection observation. This result shows that, reliably estimating specular reflectance for all surface points is still difficult unless there are enough observation of specular reflections for every surface point.



(a) input image

(b) ground-truth diffuse image

(c) ground-truth specular image

(d) ground-truth shading

(e) initial shape

(f) synthesized image

(g) synthesized diffuse image

(h) synthesized specular image

(i) estimated shading

Fig. 9. Result for the "amphora" image set (36 images) — textured non-Lambertian surface case (spatially varying specular reflectance, static illumination, and varying viewpoint). 95% accuracy (shape, ρ_{dr}, ρ_{dg}, ρ_{db}, ρ_s, α_s)=(0.59mm, 0.041, 0.047, 0.042, 0.226, 12.69), 1.0mm completeness (shape, ρ_{dr}, ρ_{dg}, ρ_{db}, ρ_s, α_s) = (89.73%, 0.042, 0.047, 0.042, 0.226, 12.65).

3 Conclusion

In this paper, we have given a coarse overview of our works on multi-view 3D and appearance modeling. Contrary to previous works that consider specific scenarios, our approach can be applied indiscriminately to a number of classical scenarios — it naturally fuses and exploits several important cues (silhouettes, stereo, and shading) and allows to deal with most of the classical 3D reconstruction scenarios such as stereo vision, (multi-view) photometric stereo, and multi-view shape from shading. In addition, our method can deal with non-Lambertian surfaces showing strong specular reflection, which is difficult even in some other state of the art methods using complex similarity measures. Technical details are given in our previous publications. Also, although the proposed approach can in principle deal with very general scenarios, especially the case of estimating specular coefficients remains challenging in practice due to numerical issues. A discussion of such practical aspects is provided in [9].

Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2006-352-D00087) and by the FLAMENCO project (grant ANR-06-MDCA-007).

References

1. Birkbeck, N., Cobzas, D., Sturm, P., Jägersand, M.: Variational shape and reflectance estimation under changing light and viewpoints. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 536–549. Springer, Heidelberg (2006)
2. Delaunoy, A., Gargallo, P., Prados, E., Pons, J.-P., Sturm, P.: Minimizing the multiview stereo reprojection error for triangular surface meshes. In: British Machine Vision Conference (2008)
3. Gargallo, P., Prados, E., Sturm, P.: Minimizing the reprojection error in surface reconstruction from images. In: IEEE International Conference on Computer Vision (2007)
4. Hernández Esteban, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 548–554 (2008)
5. Pons, J.-P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. International Journal of Computer Vision 72(2), 179–193 (2007)
6. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 519–528 (2006)
7. Yezzi, A., Soatto, S.: Stereoscopic segmentation. International Journal of Computer Vision 53(1), 31–43 (2003)
8. Yoon, K.-J., Prados, E., Sturm, P.: Generic scene recovery using multiple images. In: International Conference on Scale Space and Variational Methods in Computer Vision (2009)
9. Yoon, K.-J., Prados, E., Sturm, P.: Joint estimation of shape and reflectance using multiple images with known illumination conditions. International Journal of Computer Vision (to appear, 2009)

XIII Computer Vision

Towards an Iterative Algorithm for the Optimal Boundary Coverage of a 3D Environment

Andrea Bottino

Politecnico di Torino, Dipartimento di Automatica e Informatica,
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
andrea.bottino@polito.it

Abstract. This paper presents a new optimal algorithm for locating a set of sensors in 3D able to see the boundaries of a polyhedral environment. Our approach is iterative and is based on a lower bound on the sensors' number and on a restriction of the original problem requiring each face to be observed in its entirety by at least one sensor. The lower bound allows evaluating the quality of the solution obtained at each step, and halting the algorithm if the solution is satisfactory. The algorithm asymptotically converges to the optimal solution of the unrestricted problem if the faces are subdivided into smaller parts.

Keywords: 3D sensor positioning, Art Gallery, lower bound.

1 Introduction

Sensor planning is an important research area in computer vision. It consists of automatically computing sensor positions or trajectories given a task to perform, the sensor features and a model of the environment. A recent survey [2] refers in particular to tasks as reconstruction and inspection. Several other tasks and techniques were considered in the more seasoned surveys [3] and [4].

Sensor panning problems require considering a number of constraints, first of all visibility. To this effect, the sensor is usually modeled as a point and referred to as a "viewpoint". A feature of an object is said to be visible from the viewpoint if any segment joining a point of the feature and the viewpoint does not intersects the environment or the object itself (usually excluding boundary points). Assuming omnidirectional or rotating sensors, for tasks such as surveillance, the visibility constraint is modeled by the classic Art Gallery problem, which requires observing or "covering" the interior of a polyhedral environment with a minimum set of sensors. We call this the Interior Covering problem (IC). The problem tackled in this paper is similar, but not identical. It requires observing only the boundaries of P, faces for a polyhedral environment, and it applies for instance in problems like inspection or image based rendering. We call this the Face Covering (FC) problem.

FC and IC problems are NP-hard. However, "good" approximation algorithms are sorely needed. In our view, a "good" practical algorithm should not only be computationally feasible, but also provide a set of sensors whose cardinality, on the average, is not far from optimum. In this paper, we present a new FC sensor positioning technique. The algorithm is incremental and converges toward the optimal solution. A key feature of the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 707–715, 2009.

algorithm is that it computes a lower bound, specific of the polyhedral environment considered, for the minimum number of sensors. It allows evaluating the quality of the solution obtained at each step, and halting the algorithm if the solution is satisfactory. The algorithm refines a starting approximate solution provided by an integer face covering algorithm, (IFC) where each face must be observed entirely by at least one sensor. A set of rules, aimed to reduce the computation, is provided for refining locally the current solution.

Compared to the large amount of literature on the 2D case, relatively few articles on 3D sensor positioning have been published. Furthermore, to our knowledge, currently no method for automatic sensor planning in a 3D polyhedral environment, providing an evaluation of the coverage and information for improving it towards the optimum has been presented. Tarabanis presents in [6] an algorithm for computing the locus of viewpoints from which faces of polyhedral objects are seen in their entirety. These can be used solve the FC problems, but no indication where to locate a minimal number of sensor for seeing all the features of the object is provided. Rana [7] presents a 2D approach that can be applied to 3D as well, where two heuristics are presented for solving the boundary coverage problem. Again, non indication on optimality of the solution is given. In [5] a graph representation is used to group faces that satisfy all constraints simultaneously and hence are suitable for viewing from a common viewpoint. The set covering problem is solved with a greedy algorithm. The only attempt to define a quality measure for the covering is given in [8], where such measure is used to compute the minimal number of 3D sensor able to cover a polyhedral environment. However, sensor position is restricted to lie on the tessellated boundaries of a reduced area, the walking zone. To avoid the case of faces of the environment not covered entirely by one sensor, all "big faces" are initially split into smaller ones. However, no indications are given to when a face must be subdivided and no certainty of the fact that all sub-faces are visible from the same sensor can be given.

2 Outline of the Algorithm

The algorithm aims at finding an optimal boundary cover of an environment P that is assumed to consist of polyhedra (with or without holes). Both internal and external coverage of the environment are managed. We stress that our work is focused on the optimality of the solutions provided by the algorithm. The approach is incremental, and it starts from an initial solution which is refined step by step. The initial step is given by a useful reduction of the covering problem, the Integer Face Covering (IFC), where each face must be covered in its entirety by at least one sensor. This (restricted) problem has an optimal solution, provided by the Integer Face Covering Algorithm (IFCA). In order to develop an effective incremental algorithm, it is also necessary to have a technique for evaluating at each step the quality of the current approximate solution, and an algorithm able to refine locally the solution, in order to reduce the computational burden and leading towards the optimum. A key component for the first step is the evaluation of a lower bound LB(P) on the number of sensors that is specific to the polyhedron P considered. Its value can be compared with the current solution and, if the result is not satisfactory, this can be refined by *dividing* some of the faces of P into smaller areas and applying again IFCA. For this task, the INDIVA$_{3D}$ algorithm allows finding the faces of P that must not be split (that is, the "*indivisible*" faces) since they are entirely observed by at least one guard of all optimal solutions.

The outline of the incremental algorithm is as follows:

- ***Step 1***. Compute a lower bound LB(P), specific to the polyhedron P, for the cardinality of the minimum set of guards using the algorithm LBA_{3D}.
- ***Step 2***. Compute an integer face cover of cardinality IFCC using the algorithm IFCA.
- ***Step 3***. Compare LB(P) and IFCC. If they are equal, or the relative maximum error (IFCC-LB(P))/LB(P) is less than a predefined threshold, ***STOP***. Otherwise:
- ***Step 4***. Apply algorithm $INDIVA_{3D}$ for finding indivisible faces. If all faces are indivisible, ***STOP***, since IFC is optimal. Otherwise, split the remaining faces and compute a new lower bound.
- Compare the new lower bound and the current IFCC. If they are equal, ***STOP***. Otherwise go to ***Step 2***.

For a practical implementation, the algorithm can be halted if several consecutive steps have not changed the cardinality of the current solution. Clearly, the algorithm converges toward an optimal solution in an undefined number of steps. In the following paragraphs, we will detail the basic components of the algorithm.

2.1 Integer Face Covering Algorithm (IFCA)

Integer face covering (IFC) requires each face to be entirely covered by at least one guard. First, let us observe one fact. Let the *Integer Visibility Region* I(f) of a face f be the region of the viewing space whose points see entirely f. An IFC cover requires a sensor to be placed in the I(f) of every face of P. However, while a non empty I(f) exists for every convex face, this is not true in the case of concave faces. This can be seen from the example in Fig. 1(a), where, considering internal covering of P, I(f_1) and I(f_2) are empty. Therefore, in order to guarantee the IFC problem has a solution, we require that any concave face is initially split into convex parts, as in Fig. 1 (b).

Given this initial constraint, a simple example showing the difference between FC and IFC is shown in Fig. 2, where three sensors are necessary for the integer covering of the faces of polyhedron P (a), while only two FC sensors are necessary (b).

Regarding complexity, a detailed analysis is omitted for the sake of conciseness, but is similar to the one presented in [1]. The relevant point is that IFC is NP-complete and finite algorithms are possible [9]. An algorithm of this kind, working for any polyhedral environment (external coverage of multiple polygons, internal coverage of polygons with or without holes) is described and implemented in [9]. Here we will present only the main lines of this algorithm, which are necessary for fully understanding its incremental extension. The steps of the IFC algorithm are the following:

IFCA

Step 1. Compute a partition Π of the viewing space into regions Zi such that:

- The same set $\boldsymbol{F}_i$=(f_p, f_q,…, f_t) of faces is entirely visible from each point of Z_i, $\forall i$
- The regions Z_i are maximal regions, that is $\boldsymbol{F_i} \not\subset \boldsymbol{F_j}$ where Z_j is any region bordering Z_i

Step 2. Select the dominant regions and the essential regions. A region Z_i is dominant if there is no other region Z_j such that $\boldsymbol{F_i} \subset \boldsymbol{F_j}$. An essential zone is a dominant zone that covers a face not covered by any other dominant zone.

Step 3. Select an optimal (or minimal) solution. A minimal solution consists of a set containing all the essential and some dominant regions $\mathbf{S_j}$ = (Z_{j1}, Z_{j2},..., Z_{jk}) such that it covers all faces with the minimum number of members.

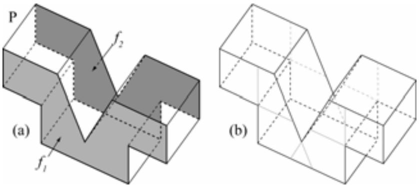


Fig. 1. In the case of internal covering, I(f_1) and I(f_2) are empty (a); a convex decomposition of the two faces (b)

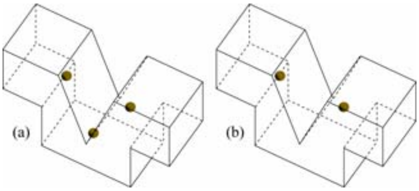


Fig. 2. Three IFC sensors are required (a) while only two FC sensors are necessary (b)

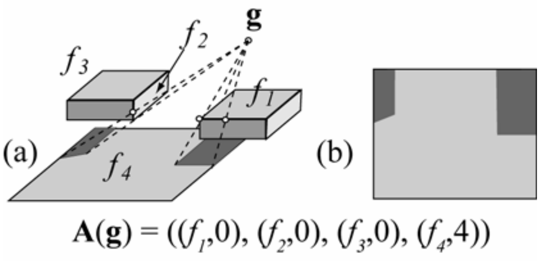


Fig. 3. An example of aspect of point **g** (a). The projection of the four occluding features (edges) of P on f (b).

The main difference of the current approach with the algorithm in [9] is how Π is built. Here a two phase process is necessary. In the first phase, a more detailed partition Π', whose regions are also used by LBA$_{3D}$ and INDIVA$_{3D}$, is constructed. In the second phase, Π' is refined to obtain Π.

Before defining Π', let us define the *aspect* **A**(**g**) of a point **g**:

A(**g**) = ((f_h, n_h), (f_k, n_k), … (f_q, n_q))

where f_h, f_k, … f_q are the faces fully or partially visible from **g**, and n_h, n_k, … n_q are the number of occlusions for each face. The number of occlusions is, briefly, the number of edges of P that are entirely or partially projected on f from **g**. The aspect defines if a face f is partially visible ($n_f \neq 0$), totally visible ($n_f = 0$) or not visible (f not in the aspect) from **g**. The word aspect has been used in agreement with the literature on aspect graphs. The interested reader is referred to the survey paper [10]. An example of aspect is shown in Fig. 3.

Partition Π' is defined as the partition that divides the interior/exterior of P into regions Z'_i such that

- All points of Z'_i have the same aspect $\mathbf{A_i}$
- Z'_i are maximum regions, i.e., $\mathbf{A_i} \neq \mathbf{A_j}$ for contiguous regions.

The construction of Π'can be performed using a set of *active patches*, belonging to *active surfaces*. The active patches are the boundaries between points whose aspects are different. The active patches are a subset of four kinds of active surfaces related to a face f of P:

- ***Type I***: the plane supporting f
- ***Type II***: surfaces originating from a vertex of f and tangent to an edge of P (VE surfaces), or from an edge of f and tangent to a vertex of P (EV surfaces)

- ***Type III***: EEE surfaces (ruled surfaces), tangent to an edge *e* of *f* and two other edges of P, or tangent to three edges of P and intersecting *f*
- ***Type IV:*** planar surfaces tangent to two parallel edges of P and intersecting *f*

According to the geometry of these surfaces, to each active surface can be associated one or more active patches, and to each patch a particular *visual event*. A visual event is a rule for changing the aspect of an imaginary viewpoint crossing the active patch, and it is synthesized by a *3D cross operator* having a positive and a negative direction. The *positive visual event* is the change of aspect of a point crossing the active patch along the positive direction; a similar definition holds for the negative visual event. Therefore, after constructing the partition Π' using the active patches, the aspect of each region can be constructed with a visiting algorithm, starting from a region whose aspect has been computed. The complete catalogue of active surfaces and active patches is shown in Fig. 4. The changes in the aspect due to the different 3D cross operators are listed in Table 1. A further analysis of the active patches might be necessary since, crossing an active patch T, *f* can be partially or totally hidden by other parts of the polyhedron P not related to the feature originating T. A detailed analysis of the different cases will not be performed here, for the sake of brevity.

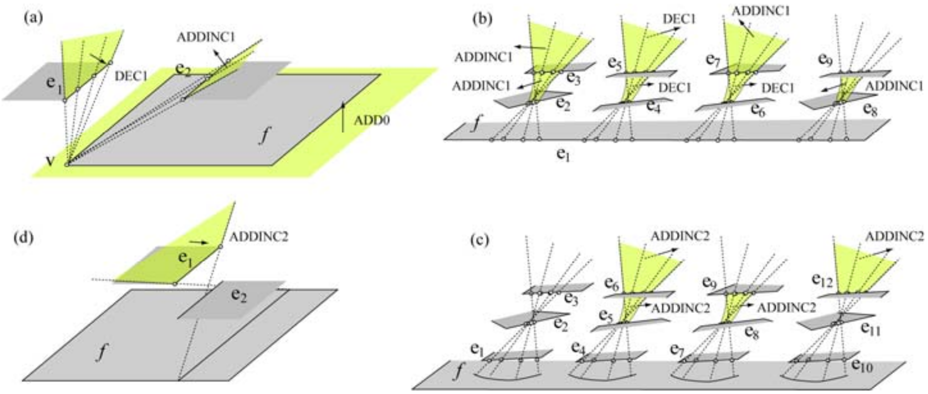


Fig. 4. The catalogue of active surfaces related to a face *f*. a) type I and type II surfaces. b) and c) type III. d) type IV. For each surface, the active patches are highlighted, together with their associated cross operator.

Table 1. Cross operators and corresponding positive and negative visual events

3D Cross Operator	Positive visual event	Negative visual event
ADD0(f_i)	Add (f_i,*0*) to aspect	Delete (f_i, *0*) from aspect
DEC1(f_i)	$n_i = n_i - 1$	$n_i = n_i + 1$

ADDINC*k*(f_i) ***k*=[1,2]**	if(f_i in aspect) → $n_i = n_i + k$
	else → Add (f_i, k)
		if($n_i == k$) → Delete (f_i, k)
		else → $n_i = n_i - k$

2.2 Lower Bound Algorithm (LBA3D)

LBA$_{3D}$ computes a lower bound of the number of sensors, specific to P, for the unrestricted sensor positioning problem. Both LBA$_{3D}$ and INDIVA$_{3D}$ algorithms make use of the concept of weak and integer visibility regions of a face. They can be defined as follows:

- the ***Weak visibility region*** W(f_i) of a face f_i is the 3D region whose points see at least a point of f_i; points seeing only boundaries of f_i do not belong to W(f_i)
- the ***Integer visibility region*** I(f_i) is the 3D region whose points see entirely f_i

An example of weak and integer visibility regions for a face *f* of a simple polyhedron can be seen in Fig. 5.

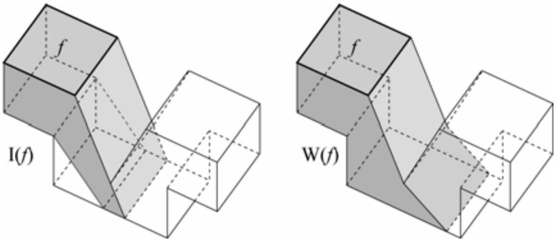


Fig. 5. Integer and weak visibility regions of face *f*

Both regions can be easily obtained as a byproduct of IFCA. In fact, given the aspect of each region of Π', W(f_i) and I(f_i) are simply obtained by merging the regions whose aspect contains f_i, in the former case, or where the number of occlusions of f_i is zero, in the latter case. If there are *p* zones in the partition Π, and *n* faces in P, computing the visibility regions for all faces is O(*pn*).

Weak visibility regions allow us to determine a lower bound for the number of sensors needed. It is easy to see that:

Statement 1: *The cardinality of the maximal subset of disjoint (not intersecting) weak visibility regions W(f_i) of P is a lower bound LB(P) for the minimal number of sensors*

In fact, since each weak visibility region must contain at least one sensor, no arrangement of face covering sensors can have fewer sensors than LB(P).

Computing the lower bound requires solving the *maximum independent set* problem for a graph G where each node represents the weak visibility region of a face of P and each face of G connects nodes corresponding to intersecting visibility regions. The problem is equivalent to the *maximum clique problem* for the *complement graph* G' (the graph obtained by joining those pairs of vertices that are not adjacent in G). It is well known that these are again NP-complete problems, but *exact* branch-and-bound algorithms for these problems have been presented and extensively tested ([11], [12], [13]), showing more than acceptable performances for graphs with hundreds of nodes. Then, computing LB(P) is computationally feasible for practical cases.

2.3 INDIVisible Faces Algorithm (INDIVA3D)

If optimal sets of sensors exist such that a face is entirely observed by at least one sensor of each set, then, in order to approach these optimal solutions, that face does not need to be split. Such a face is called *indivisible*. The rules for finding the indivisible faces of P are as follows:

Rule1. If W(f_i) = I(f_i), f_i is indivisible.

Rule2. If W(f_i) $\subseteq$ I(f_j), f_j is indivisible.

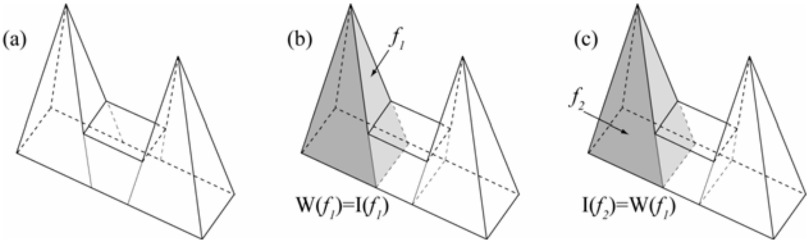


Fig. 6. In this case, the optimum FC and IFC covers are equal

Both rules follow from the fact that, for any solution, *at least one sensor of any minimal set* must be located in each weak visibility region. If the weak region is equal to the integer region of the face, rule 1, then for any solution the sensor placed in the weak region observes the face in its entirety. If the weak region is included in the integer region of another face, rule 2, then the sensor placed in the weak region observes the second region in its entirety. It follows that for every solution (and in particular for every optimal solution) these faces are observed in their entirety by at least one sensor and, therefore, they do not need to be divided.

A simple example will show how to apply these rules, and that they are powerful tools for simplifying the problem. Let us consider the polyhedron shown in Fig. 6(a) with the subdivision of its concave faces. The integer and weak visibility regions of face f_1 are coincident, and therefore for rule 1 the face is indivisible. The integer visibility region of f_2 is equal to the weak visibility region of f_1, and therefore is indivisible for rule 2. It can be easily seen that all the faces of P are indivisible, and then the unrestricted minimal set of guards is that provided by IFCA. The same result could have been obtained by computing the IFC solution, whose cardinality is equal to the lower bound LB(P).

Divisible faces must be partitioned, for instance, by splitting in two all the edges of the face, and connecting the central point of each edge with the face center.

2.4 Examples

A first example of how the algorithm works can be seen in Fig. 7. In (a) the polyhedron P is shown. LB(P) is 2, and in (b) the two not-intersecting weak polygons of faces f_1 and f_2 are shown, together with the initial solution of IFCA, whose cardinality is three. Applying rules 1-2, face f_3 is found to be the only divisible face and (c) shows its decomposition. Applying again IFCA, we obtain a solution with only two sensors (d), whose cardinality is equal to LB(P) and therefore is optimal.

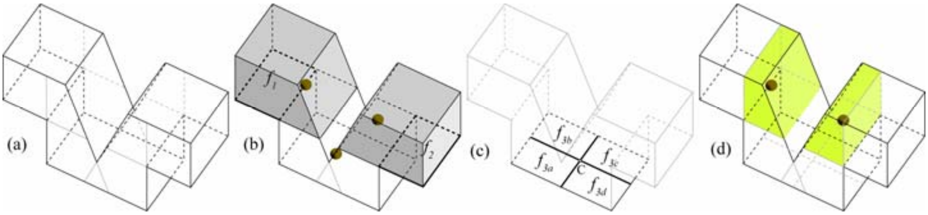


Fig. 7. Polyhedron P (a), W(f_1) and W(f_2), determining LB(P) = 2, and the initial IFC covering of cardinality three (b), subdivision of f_3 (c), the solution of the second iteration of IFCA and the regions where sensors can be located (d)

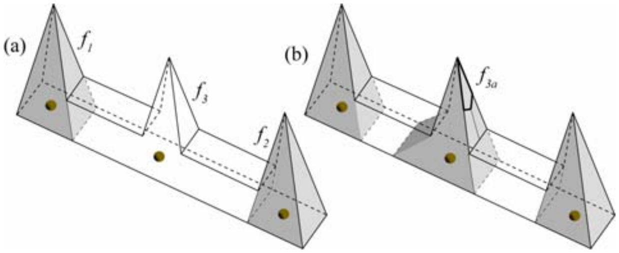


Fig. 8. Dividing faces of the initial polyhedron in (a) the lower bound can increase (b)

Observe that the lower bound is evaluated at every iteration. In fact, it might increase, and then improve, after splitting some faces of the polygon. An example can be seen in Fig. 8. In (a) the comb polyhedron is shown, together with the initial lower bound, whose value is 2, the non-intersecting weak regions of faces f_1 and f_2, and an initial IFCA solution, whose cardinality is three. Face f_3 is found to be divisible, and is split. Now, consider face f_{3a} drawn in bold in (b). Its weak polygon, also shown in the picture, does not intersect W(f_1) and W(f_2) and the value of the lower bound increases to three. The new lower bound is equal to the cardinality of the previous IFCA solution that is, therefore, optimal.

3 Conclusions

This paper presents an incremental algorithm for positioning sensors in a 3D environment that are capable of seeing in their entirety the external or internal boundaries of a polyhedral environment. The approach is iterative and it is based on a lower bound on the number of sensor that allows to evaluate the closeness to optimality of the solution and to define rules for trying to improve the current solution. This is, in our knowledge, the first work in literature that attempts to tackle this problem in 3D. Future work will be focused on the full implementation of the algorithm, which is a rather complex task, especially for the generation and intersection of the active patches, and to extend it to take into account several additional constraints besides the visibility one.

References

[1] Bottino, A., Laurentini, A.: A Nearly Optimal Sensor Placement Algorithm for Boundary Coverage. Pattern Recognition 41(11), 3343–3355 (2008)

[2] Scott, W.R., Roth, G.: View Planning for Automated Three-Dimensional Object Reconstruction and Inspection. ACM Computing Surveys 35(1), 64–96 (2003)

[3] Tarabanis, K.A., Allen, P.K., Tsai, R.Y.: A survey of sensor planning in computer vision. IEEE Trans. Robot. and Automat. 11(1), 86–104 (1995)

[4] Newman, T.S., Jain, A.K.: A survey of automated visual inspection. Comput. Vis. Image Understand. 61(2), 231–262 (1995)

[5] Roberts, D.R., Marshall, A.D.: Viewpoint Selection for Complete Surface Coverage of Three Dimensional Objects. In: Proc.of the Britsh Machine Vision Conference (1998)

[6] Tarabanis, K., Tsai, R.Y., Kaul, A.: Computing occlusion-free viewpoints. IEEE Trans. Pattern Analysis and Machine Intelligence 18(3), 279–292 (1996)

[7] Rana, S.: Two approximate solutions to the Art Gallery Problem. In: Proc. of International Conference on Computer Graphics and Interactive Techniques (2004)

[8] Fleishman, S., Cohen-Or, D., Lischinski, D.: Automatic Camera Placement for Image-Based Modeling. In: Proc. 7th Pacific Conf. on CG and Applications, pp. 12–20 (1999)

[9] Bottino, A., Laurentini, A.: Optimal positioning of sensors in 3D. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 804–812. Springer, Heidelberg (2005)

[10] Schiffenbauer, R.: A Survey of Aspect Graphs. Polytechnic University, Brooklyn, Technical Report TR-CIS-2001-01 (2001)

[11] Woods, D.R.: An algorithm for finding a maximum clique in a graph. Operations Research Letters 21, 211–217 (1997)

[12] Carraghan, D.R., Paradalos, P.: An exact algorithm for the maximum clique problem. Operations Research Letters 9, 375–382 (1990)

[13] Oestergard, P.: A fast algorithm for the maximum clique problem. Discrete Applied Mathematics 120, 197–207 (2002)

[14] Fleishman, S., Cohen-Or, D., Lischinski, D.: Automatic Camera Placement for Image-Based Modeling. In: Proc. Of 7th Pacific conf. on Computer Graphics and Applications (1999)

Measuring Cubeness of 3D Shapes

Carlos Martinez-Ortiz and Joviša Žunić*

Department of Computer Science, University of Exeter, Exeter EX4 4QF, U.K.
{cm265,J.Zunic}@ex.ac.uk

Abstract. In this paper we introduce a new measure for $3D$ shapes: cubeness. The new measure ranges over $(0, 1]$ and reaches 1 only when the given shapes is a cube. The new measure is invariant with respect to rotation, translation and scaling, and is also robust with respect to noise.

Keywords: $3D$ shape, compactness measure, image processing.

1 Introduction

Shape descriptors are a powerful tool for shape classification tasks in $2D$ and $3D$. Many shape descriptors are already studied in the literature and used in practice. Some of the best known ones in $2D$ are: elongation([10]), convexity([5]), rectangularity([6]), rectilinearity([13]), sigmoidality([7]), circularity([12]), etc. There are also some $3D$ shape descriptors like: compactness([1,2]), geometric moments ([4]), Fourier Transforms ([11]), etc.

In this paper we define a new $3D$ shape descriptor which measures the similarity of an object and a cube. We call this new measure "cubeness". Notice that the $3D$ measure $\mathcal{C}_d(S)$, presented in [2], is similar in some respect to the measure introduced here: it is maximised by a cube – i.e. $\mathcal{C}_d(S)$ picks up the highest possible value (which is 1) if and only if the measured shape is cube. Such measure is defined as follows:

$$\mathcal{C}_d(S) = \frac{n(S) - A(S)/6}{n - (\sqrt[3]{n(S)})^2} \tag{1}$$

where $A(S)$ is the area of the enclosing surface, i.e. the sum of the area of the voxels faces which form the surface of the shape, and $n(S)$ is the number of voxels in the shape.

Measure $\mathcal{C}_d(S)$ is a measure of discrete compactness of rigid solids composed of a finite number of polyhedrons. When these polyhedrons are voxels, the most compact shape according to $\mathcal{C}_d(S)$ is a cube and thus it was select as a comparable measure to the measure introduced in this paper.

One possible application of the new cubeness measure introduced in this paper can be as an additional feature for $3D$ search engines like the one presented in [3]. Their search engine uses spherical harmonics to compute similarity measures

[*] J. Žunić is also with the Mathematical Institute, Serbian Academy of Arts and Sciences, Belgrade.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 716–723, 2009.

used for the search. The cubeness measure presented in this paper could be used as an additional similarity measure for such a search engine. Some experiments on this point will be performed in the future.

This paper is organised as follows. The next section introduces the new cubeness measure and highlights several of its desirable properties. Section 3 gives several examples which demonstrate the behaviour of the new measure. Section 4 contains some comments and conclusions.

2 Cubeness Measure

In this section we define the new cubeness measure. Throughout this paper we will assume that all appearing shapes have non-empty interior, i.e. they have a strictly positive volume. We will also assume that two shapes S_1 and S_2 to be equal if the symmetric set difference $(S_1 \setminus S_2) \cup (S_2 \setminus S_1)$ has volume zero. Such assumptions are necessary to keep the proofs mathematically rigorous, but they are not of practical importance – e.g. under these assumptions the open ball $\{(x,y,z) \mid x^2+y^2+z^2<1\}$ and closed one $\{(x,y,z) \mid x^2+y^2+z^2 \leq 1\}$ are the same shape which is totally acceptable from the view point of image processing and computer vision applications, even that they differ for a spherical surface $\{(x,y,z) \mid x^2+y^2+z^2=1\}$ (having the volume equal to zero). Also any appearing shape will be considered that its centroid coincides with the origin even if not explicitly stated. $S(\alpha,\beta)$ will denote the shape S rotated along the X axis by an angle α and, along the Y axis by an angle β. We will use the l_∞-distance in our derivation; just a short remainder that l_∞-distance between points $A=(a_1,a_2,a_3)$ and $B=(b_1,b_2,b_3)$ is defined as:

$$l_\infty(A,B) = max\{|a_1-b_1|, |a_2-b_2|, |a_3-b_3|\}. \tag{2}$$

Trivially, the set of all points $X=(x,y,z)$ whose l_∞-distance from the origin $O=(0,0,0)$ is not bigger than r is a cube. Such a cube will be denoted by $\mathcal{Q}(r)$:

$$\mathcal{Q}(r) = \{X=(x,y,z) \mid l_\infty(X,O) \leq r\} = \{(x,y,z) \mid \max\{|x|,|y|,|z|\} \leq r\}. \tag{3}$$

To define the new cubeness measure, we start with the quantity:

$$\min_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\} dxdydz \tag{4}$$

and show that such a quantity reaches its minimum value if and only if the shape S is a cube. By exploiting this nice property we will come to a new cubeness measure. First we prove the following theorem:

Theorem 1. *Let S be a given shape whose centroid coincides with the origin, and let $S(\alpha,\beta)$ denote the shape S rotated along the X axis by an angle α and along the Y axis by an angle β. Then,*

$$\frac{\iiint\limits_S \max\{|x|,|y|,|z|\}dxdydz}{Volume(S)^{4/3}} \geq \frac{3}{8} \tag{5}$$

$$\frac{\iiint\limits_S \max\{|x|,|y|,|z|\}dxdydz}{Volume(S)^{4/3}} = \frac{3}{8} \iff S = \mathcal{Q}\left(\frac{Volume(S)^{1/3}}{2}\right) \tag{6}$$

$$\frac{\min\limits_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} max\{|x|,|y|,|z|\}dxdydz}{Volume(S)^{4/3}} = \frac{3}{8} \iff S \ is \ a \ cube. \tag{7}$$

Proof. Let S be a shape as in the statement of the theorem. Also, let $\mathcal{Q}$, for short, denote the cube $\mathcal{Q}(\frac{Volume(S)^{1/3}}{2})$, i.e. the cube is aligned with the coordinate axes and the faces intersect the axes at points: $(a/2,0,0)$, $(-a/2,0,0)$, $(0,a/2,0)$, $(0,-a/2,0)$, $(0,0,a/2)$ and $(0,0,-a/2)$ and $a = Volume(S)^{1/3}$ (see Fig.1(a)). Trivially, the volumes of S and $\mathcal{Q}$ are the same, and also:

(i) The volume of the set differences $S \setminus \mathcal{Q}$ and $\mathcal{Q} \setminus S$ are the same, because the volumes of S and $\mathcal{Q}$ are the same;

(ii) The points from $\mathcal{Q} \setminus S$ are closer (with respect to l_∞-distance) to the origin than the points from $S \setminus \mathcal{Q}$. More formally: if $(u,v,w) \in S \setminus \mathcal{Q}$ and $(p,q,r) \in \mathcal{Q} \setminus S$, then $\max\{|u|,|v|,|w|\} > \max\{|p|,|q|,|r|\}$ (see. Fig. 1 (b) and (c)).

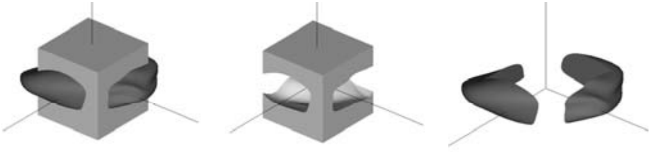

(a) $\mathcal{Q}$ and S (b) $\mathcal{Q} \setminus S$ (c) $S \setminus \mathcal{Q}$

Fig. 1. Shapes S and $\mathcal{Q} = \mathcal{Q}\left(Volume(S)^{1/3}/2\right)$. Both shapes have the same volume. Points in $\mathcal{Q} \setminus S$ are closer to the origin (using l_∞-distance) than those in $S \setminus \mathcal{Q}$.

Further (i) and (ii) give:

$$\iiint\limits_{S\setminus\mathcal{Q}} \max\{|x|,|y|,|z|\}dxdydz \geq \iiint\limits_{\mathcal{Q}\setminus S} \max\{|x|,|y|,|z|\}dxdydz. \tag{8}$$

Now, we derive:

$$\iiint\limits_S \max\{|x|,|y|,|z|\}dxdydz = 8 \iiint\limits_{\substack{(x,y,z)\in S\\ x,y,z\geq 0}} \max\{x,y,z\}dxdydz$$

$$= 48 \iiint\limits_{\substack{(x,y,z)\in S\\ x\geq y\geq z\geq 0}} xdxdydz = 48 \int\limits_0^{a/2}\int\limits_0^{x}\int\limits_0^{y} xdxdydz = \frac{3}{8}\cdot a^4,$$

which proves (5) since $a = Volume(S)^{1/3}$.

The proof of (6) comes from the fact that equality (8) holds only when the shapes S and Q are the same, i.e. when $Volume(S \setminus \mathcal{Q}) = Volume(\mathcal{Q} \setminus S) = 0$.

To prove (7) let α_0 and β_0 be the angles which minimise $\iiint\limits_S max\{|x|,|y|,|z|\}dxdydz$:

$$\iiint\limits_{S(\alpha_0,\beta_0)} max\{|x|,|y|,|z|\}dxdydz = \min_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} max\{|x|,|y|,|z|\}. \tag{9}$$

Since $Volume(S) = Volume(S(\alpha,\beta)) = Volume(S(\alpha_0,\beta_0))$, then (see (6))

$$\frac{\iiint\limits_{S(\alpha_0,\beta_0)} \max\{|x|,|y|,|z|\}dxdydz}{Volume(S(\alpha_0,\beta_0))^{4/3}} = \frac{3}{8}$$

would imply that $S(\alpha_0,\beta_0)$ must be equal to $\mathcal{Q}$ – i.e., S must be a cube. □

Theorem 1 tells us that $\iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz$ reaches its minimum value of 3/8 only when S is a cube. Based on this, we give the following definition for the cubeness measure.

Definition 1. *The cubeness measure $\mathcal{C}(S)$ of a given shape S is defined as*

$$\mathcal{C}(S) = \frac{3}{8} \cdot \frac{Volume(S)^{4/3}}{\min\limits_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz}. \tag{10}$$

The following theorem summarizes the desirable properties of $\mathcal{C}(S)$.

Theorem 2. *The cubeness measure $\mathcal{C}(S)$ has the following properties:*

(a) $\mathcal{C}(S) \in (0,1]$, for all 3D shapes S with non-empty interior;
(b) $\mathcal{C}(S) = 1 \iff S$ is a cube;
(c) $\mathcal{C}(S)$ is invariant with respect to similarity transformations;

Proof. Items (a) and (b) follow directly from Theorem 1.

Item (c) follows from the fact that both $\min\limits_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz$ and volume of S are rotation invariant, which makes $\mathcal{C}(S)$ rotation invariant. $\mathcal{C}(S)$ is translation invariant by definition, since it is assumed that the centroid of S coincides with the origin. Finally if S is scaled by a factor of $\mathbf{r}$ then easily

$$\min_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{\mathbf{r}\cdot S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz$$

$$= \mathbf{r}^4 \cdot \min_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz$$

and

$$Volume(\mathbf{r} \cdot S) = \mathbf{r}^3 \cdot Volume(S)$$

and, consequently,

$$\begin{aligned}\mathcal{C}(\mathbf{r} \cdot S) &= \frac{3}{8} \cdot \frac{Volume(\mathbf{r} \cdot S)^{4/3}}{\min\limits_{\alpha \in [0,2\pi], \beta \in [0,2\pi]} \iiint\limits_{\mathbf{r} \cdot S(\alpha,\beta)} \max\{|x|,|y|,|z|\} dxdydz} \\ &= \frac{3}{8} \cdot \frac{(\mathbf{r}^3 \cdot Volume(S))^{4/3}}{\mathbf{r}^4 \cdot \min\limits_{\alpha \in [0,2\pi], \beta \in [0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\} dxdydz} \quad = \quad \mathcal{C}(S)\end{aligned}$$

which means that $\mathcal{C}(S)$ is scale invariant. □

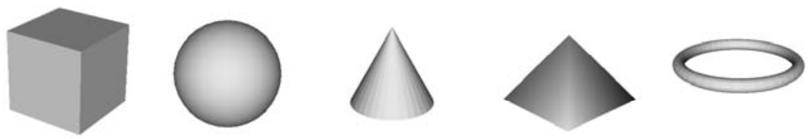
The cubeness measure $\mathcal{C}(S)$ is very similar in spirit to the compactness measure presented in [14]:

$$\mathcal{K}(S) = \frac{3^{5/3}}{5(4\pi)^{2/3}} \cdot \frac{\mu_{0,0,0}(S)^{5/3}}{\mu_{2,0,0}(S) + \mu_{0,2,0}(S) + \mu_{0,0,2}(S)}.$$

however they differ in the distance measure used: $\mathcal{K}(S)$ uses euclidean distance while $\mathcal{C}(S)$ uses l_∞-distance. Therefore $\mathcal{C}(S)$ is indeed a form of compactness maximized by a cube. $\mathcal{C}_d(S)$ is also a compactness measure maximized by a cube, while a sphere is the most compact shape according to $\mathcal{K}(S)$.

3 Experiments Illustrating $\mathcal{C}(S)$ Measure

In this section we give several examples in order to illustrate the behaviour of $\mathcal{C}(S)$. Figure 2 shows several geometric shapes ranked according to $\mathcal{C}(S)$ measure. $\mathcal{C}(S)$ is given under each figure. The values in brackets corresponds to $C_d(S)$. Notice that $C_d(S)$ measure also reaches it maximum value 1 for the cube. It is in accordance with our expectation that shapes which are more "spread out", like (e), have lower $\mathcal{C}(S)$ measure. Another fact which is worth pointing out is that the values obtained by $\mathcal{C}(S)$, for the shapes displayed, are in a wider range $[0.340 - 1.000]$ than values computed by $C_d(S)$ which are in the range



(a) 1.000 (1.000)

(b) 0.970 (0.995)

(c) 0.812 (0.734)

(d) 0.795 (0.754)

(e) 0.340 (0.785)

Fig. 2. Different geometric shapes ordered according to their cubeness $\mathcal{C}(S)$. The values $\mathcal{C}_d(S)$ are in brackets.

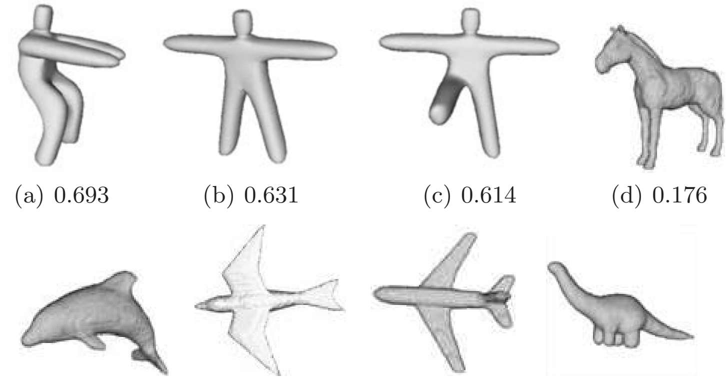
[0.734 – 1.000]. This indicates that $\mathcal{C}(S)$ assigns a more distinctive values than $\mathcal{SC}_d(S)$ measure, making their separation capacity bigger.

Figure 3 shows some examples of $3D$ shapes of different kinds of objects, ordered according to their $\mathcal{C}(S)$ measure[1]. Results are in accordance with our perception. E.g. shapes (f) and (g) are ranked very closely. This is understandable since they share some of the same features: elongated body with thin wings spread out perpendicular to the main body; therefore it would be expected that they are considered to have similar shapes; for the purpose of an image classification task, these two shapes would be very likely grouped together, which could be expected.

Figures (a), (b) and (c) are all human shapes in different positions and their measured cubeness of (b) and (c)are close to each other, however and gets a higher score because of the position of legs and arms.

Shape (h) gets a very low $\mathcal{C}(S)$ measure, due to the fact that, having a long tail and neck, the dinosaur haves a very elongated shape, which greatly differs from a cube. Compare it with the horse in (d): the horse has a much shorter neck and tail, therefore we would expect the horse to have a higher cubeness than the dinosaur, as it is the case. However the horse still has a long neck, and long and thin legs, which make his cubeness measure relatively low.

The cubeness measure $\mathcal{C}(S)$ provides some classification power which may be combined with other shape descriptors in order to correctly classify shapes. However cubeness values are not unique for any shape (with the exception of a perfect cube which will always have $\mathcal{C}(S) = 1$); similar shapes will produce similar values, but other non-similar shapes could also produce similar values. Thus it is possible, for example, to have shapes with $\mathcal{C}(S)$ scores similar to (b) but which does not look like (b). This is unavoidable and is also the reason why different shape descriptors must be combined with to achieve better classification power.



(a) 0.693 (b) 0.631 (c) 0.614 (d) 0.176

(e) 0.167 (f) 0.137 (g) 0.132 (h) 0.099

Fig. 3. Different shapes ordered according to their cubeness $\mathcal{C}(S)$

[1] Some of the shapes are taken from the McGill database: http://www.cim.mcgill.ca/ shape/benchMark/

Figure 4 shows a cube with different levels of erosion. Values of $\mathcal{C}(S)$ decrease as more and more pieces of the shape are removed. As the level of erosion increases and the shape looks less like a cube, the produced $\mathcal{C}(S)$ values drop. Small levels of erosion do not affect too much the produced values. Such a behaviour with respect to the erosion could also suggest the robustness of $\mathcal{C}(S)$ with respect to noise. Such a robustness is expected because $\mathcal{C}(S)$ is "volume based" measure – i.e. takes into account all points inside a shape.

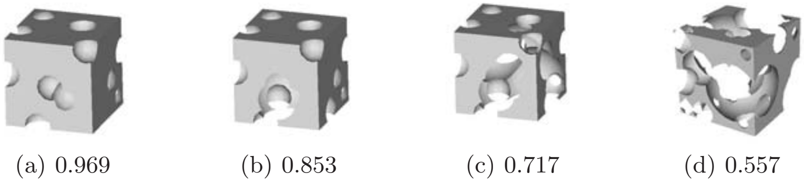
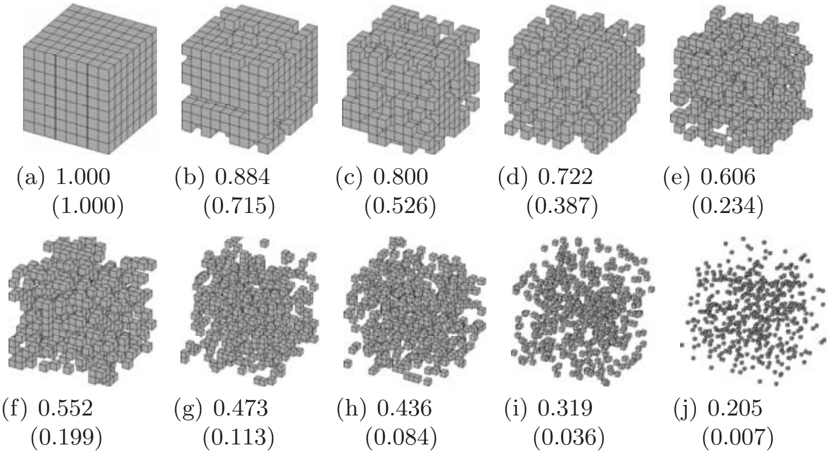

(a) 0.969 (b) 0.853 (c) 0.717 (d) 0.557

Fig. 4. Measured cubeness $\mathcal{C}(S)$ of a cube with different levels of erosion

Figure 5 illustrates some advantages that the new measure $\mathcal{C}(S)$ has over $\mathcal{C}_d(S)$. Each image displayed is composed of the same number of voxels (512) but the voxels are in different mutual positions in each image. As it can be seen $\mathcal{C}_d(S)$ decreases more rapidly, as the total surface area of all appearing components increase. This is in accordance with (1). It is easy to conclude (see (1)) that $\mathcal{C}_d(S)$ does not depend on the mutual position of such components, but just of their surfaces. On the other side, the new measure $\mathcal{C}(S)$ takes into account the mutual positions (see Definition 1 and 10). This is an advantage. Indeed, $\mathcal{C}_d(S) = 0$ holds for all shapes S where all voxels are separated, regardless of the mutual distances between them. On the other hand, $\mathcal{C}(S)$ would produce different non-zero values which vary depending on the mutual voxel positions.


(a) 1.000 (1.000) (b) 0.884 (0.715) (c) 0.800 (0.526) (d) 0.722 (0.387) (e) 0.606 (0.234)

(f) 0.552 (0.199) (g) 0.473 (0.113) (h) 0.436 (0.084) (i) 0.319 (0.036) (j) 0.205 (0.007)

Fig. 5. Same number of voxels spread over different volumes. The measured $\mathcal{C}(S)$ are immediately below the shapes while $\mathcal{C}_b(S)$ values are in brackets.

4 Conclusion

In this paper we introduce a $3D$ shape measure, cubeness, $\mathcal{C}(S)$ defined as:

$$\mathcal{C}(S) = \frac{3}{8} \cdot \frac{Volume(S)^{4/3}}{\min\limits_{\alpha,\beta\in[0,2\pi]} \iiint\limits_{S(\alpha,\beta)} \max\{|x|,|y|,|z|\}dxdydz}$$

The new measure has several desirable properties: it ranges over $(0,1]$; it gives measured cubeness of 1 if and only if the given shape is a cube; and it is invariant with respect to translation, rotation and scaling. Several experiments are given to illustrate the behaviour of the new measure. The experimental results are in accordance with theoretical considerations and with our perception. The measure works in both the discrete and continuous space, contrary to $\mathcal{C}_d(S)$ (from [2]) which is only applicable to voxelized data.

References

1. Bribiesca, E.: A measure of compactness for 3d shapes. Comput. Math. Appl. 40, 1275–1284 (2000)
2. Bribiesca, E.: An easy measure of compactness for 2d and 3d shapes. Pattern Recognition 41, 543–554 (2008)
3. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A Search Engine for 3D Models. ACM Transactions on Graphics 22(1) (2003)
4. Mamistvalov, A.G.: n-Dimensional moment invariants and conceptual mathematical theory of recognition n-dimensional solids. IEEE Trans. Patt. Anal. Mach. Intell. 20, 819–831 (1998)
5. Rahtu, E., Salo, M., Heikkila, J.: A new convexity measure based on a probabilistic interpretation of images. IEEE Trans. on Patt. Anal. and Mach. Intell. 28, 1501–1512 (2006)
6. Rosin, P.L.: Measuring shape: ellipticity, rectangularity, and triangularity. Machine Vision and Applications 14, 172–184 (2003)
7. Rosin, P.L.: Measuring sigmoidality. Patt. Rec. 37, 1735–1744 (2004)
8. Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., Dickinson, S.: Retrieving articulated 3d models using medial surfaces. Machine Vision and Applications 19(4), 261–275 (2008)
9. Sonka, M., Hlavac, V., Boyle, R.: Image Processing: Analysis and Machine Vision. In: Thomson-Engineering (September 1998)
10. Stojmenović, M., Žunić, J.: Measuring elongation from shape boundary. Journal Mathematical Imaging and Vision 30, 73–85 (2008)
11. Vranić, D.V., Saupe, D.: 3d shape descriptor based on 3d Fourier transform. In: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001), pp. 271–274 (2001)
12. Žunić, J., Hirota, K.: Measuring shape circularity. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 94–101. Springer, Heidelberg (2008)
13. Žunić, J., Rosin, P.L.: Rectilinearity measurements for polygons. IEEE Trans. on Patt. Anal. and Mach. Intell. 25, 1193–1200 (2003)
14. Žunić, J., Hirota, K., Martinez-Ortiz, C.: Compactness Measure for $3D$ Shapes (Submitted)

Self-calibration from Planes Using Differential Evolution

Luis Gerardo de la Fraga

Cinvestav, Computer Science Department,
Av. Instituto Politécnico Nacional 2508, 07360 Mexico City, Mexico
fraga@cs.cinvestav.mx

Abstract. In this work the direct self-calibration of a camera from three views of a unknown planar structure is proposed. Three views of a plane are sufficient to determine the plane structure, the view's positions and orientations and the camera's focal length. This is a non-linear optimization problem that is solved using the heuristic Differential Evolution. Once an initial structure is obtained, the bundle adjustment can be used to incorporate more views and estimate other camera intrinsic parameters and possible lens distortion. This new self-calibration method is tested with real data.

Keywords: Computer Vision, camera self-calibration, self-calibration from planes, differential evolution.

1 Introduction

Self-calibration is defined in [1, Chap. 19] as "the computation of metric properties of the cameras and/or the scene from a set of uncalibrated images". Camera (self-)calibration is one of the most important problems in computer vision. Its purpose is to obtain through a camera, an estimation of the parameters to transform a point in the real world to a point in an image. Self-calibration avoids the tedious process of calibrating cameras using special calibration objects. Using self-calibration, a camera can be calibrated on-line, i.e. every time a zoom is made. Therefore, a self-calibration technique must be used if camera's zoom is changed or auto-focus is active.

In the self-calibration process is assumed only image features correspondences to be known, and also it is possible to obtain a three-dimensional reconstruction up to an unknown similarity transformation (also called Euclidean reconstruction), but it is necessary to have some additional information about either the cameras' intrinsic parameters, the extrinsic parameters or the viewed object in order to obtain the desired Euclidean reconstruction [2,3].

Calibration using planes is a very flexible task, because it is very easy to produce high quality planar patterns (like a chessboard) easily made and printed in a laser printer. For this purpose, the Zhang [4] or Sturm [5] techniques can be used. For self-calibration with planes, no information about the planar pattern is needed, and a technique using metric rectification is available [6], but this

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 724–731, 2009.

technique needs at least four images to work. Other technique, with a different approach but also solving a global optimization task, is presented in [7].

In this paper, a novel form to solve the plane based self-calibration problem, using the heuristic Differential Evolution (DE) [8] is proposed. This new method obtains an initial plane reconstructed from three different images of that plane. DE is used to estimate the focal length and the three orientation angles and the three camera positions of each view. This is a non-linear optimization problem that is solved with DE by minimizing the reprojection error. To add more images and to correct possible camera's lens distortion, the standard bundle adjustment can be used.

The paper is organized as follows: in Sec. 2 the problem of self-calibration from planes is established. In Sec. 3 DE is briefly described. In Sec. 4 the experiment and results are shown. A briefly discussion is in Sec. 5. And finally, conclusions of this work are drawn in Sec. 6.

2 Self-calibration from Three Planes

Camera model: A point over an image is represented by $\mathbf{p} = [u, v, 1]$, a three-dimensional point is represented by a vector $\mathbf{P} = [x, y, z, 1]$. The relation between the two points is the so called pinhole camera model which is:

$$\lambda \mathbf{p} = K[R|\mathbf{t}]\mathbf{P}, \tag{1}$$

where λ is a scale factor, K a 3×3 matrix of the camera's intrinsic parameters, R a 3×3 rotation matrix and $\mathbf{t}$ a vector $[t_1, t_2, t_3]^{\mathrm{T}}$. A rotation matrix depends on only three parameters: three rotations around the main axes. The pinhole camera model in (1) represents the projection transformation of a 3D scene, by K, of a view obtained by rotating and translating the scene with respect to the world coordinates.

In the case of planes, a three-dimensional point can be also represented by the vector $\mathbf{P} = [x, y, 1]$, where the coordinate z is made equal to zero (without lost of generality, the plane is supposed to be on the xy-plane). Thus, the relation between the two points is:

$$\lambda \mathbf{p} = K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]\mathbf{P}, \tag{2}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two columns of matrix R in (1). In this work, K is defined as:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where f is the camera's focal length and (u_0, v_0) are the principal point coordinates. The principal point is the intersection point of the camera's optical axis with the image plane. The camera model in (3) assumes that pixels are squares and their xy axes are perpendicular, a supposition that can be made in modern cameras.

2.1 The Proposed Method

As input, three different views of a unknown planar pattern are needed. It is supposed that from these three images, points of correspondences have been extracted (by a method that is not part of this discussion); these points mark the position of a same characteristic (for example, corner points) viewed on the three images.

Suppose also that we have one estimation of the focal length, $\hat{f}$, the principal point, $(\hat{u}_0, \hat{v}_0)$, and the orientation, $(\hat{\theta}_1^i, \hat{\theta}_2^i, \hat{\theta}_3^i)$, and position, $(\hat{t}_1^i, \hat{t}_2^i, \hat{t}_3^i)$, for each of the three views, $i = 1, 2, 3$. Then, it is possible to calculate a model plane, this is, a set of points $\hat{\mathbf{P}}_j$ using (2), where j is equal to the number of correspondence points on the three initial images. From (2), points $\hat{\mathbf{P}}_j = [\hat{x}_j, \hat{y}_j, 1]^{\mathrm{T}}$ can be calculated by removing the scale factor λ, thus for a point (u, v) over one image, two equations as the following can be formed:

$$
\begin{aligned}
(um_{31} - m_{11})x + (um_{32} - m_{12})y &= m_{13} - um_{33} \\
(vm_{31} - m_{21})x + (vm_{32} - m_{22})y &= m_{23} - vm_{33}
\end{aligned}
$$

where m_{ij} are the elements of matrix $M = K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]$ (see Eq. (2)). Therefore, six equations can be formed, and this overdetermined system is solved using normal equations.

Here an important point: in calibration methods as [5,4] the plane model is given, in the self-calibration method the plane model is assumed to be unknown. Calibration methods are based on homographies and the self-calibration in [6] is based on inter-image homographies. An homography is the transformation mapping between two plains and it is represented by a 3×3 matrix. Homography is invariance to the scale and therefore it has only eight degrees of freedom. The proposed method here does not use homographies, therefore to impose scale invariance, the reconstructed plane is centered at its centroid and normalized arbitrarily to 0.01σ, where σ is the RMS standard deviation of all x and y values of the reconstructed model plane points.

Once the model plane is obtained and normalized, the reprojection error can be calculated as:

$$
\sum_{i=1}^{3} \sum_{j=1}^{n} \|\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}(\hat{f}, \hat{u}_0, \hat{v}_0, \hat{\theta}_1^j, \hat{\theta}_2^j, \hat{\theta}_3^j, \hat{t}_1^j, \hat{t}_2^j, \hat{t}_3^j, \hat{\mathbf{P}}_j)\|^2 \tag{4}
$$

where $\mathbf{p}_{ij}$ is the given point j, $1 \leq j \leq n$, on image i, $1 \leq i \leq 3$, and $\hat{\mathbf{p}}_i$ is the estimated point obtained from (2).

The described problem is solved with the evolutionary algorithm called Differential Evolution (DE). As an evolutionary algorithm, DE works with a population, or a set of individuals; this population evolve by mutating and selecting the best individuals, and the process stops after a given number of iterations or when other stop condition is reached. Each individual codes a possible solution for the problem, in this problem an individual is a vector of real values of size 18, which store the parameters to estimate: f, $(\theta_1^i, \theta_2^i, \theta_3^i)$ and (t_1^i, t_2^i, t_3^i), for

$i = 1, 2, 3$. The principal point (v_0, u_0), can not be estimated from three images, therefore it is fixed in the image center coordinates.

To fix the orientation of the reconstructed plane, all three image orientations are respect to $\theta_1^1 = 0$, thus this parameter is not part of an individual.

From two projections of a plane, a common rotation axis exists. The dihedral angle can be fixed if a third projection of that plane is available. This is the reason why it is necessary three images of a plane to obtain its reconstruction. This form to obtain a reconstruction is a old procedure used in computer tomography [9].

3 Differential Evolution

The population of DE is composed by a set of individuals or vectors of real numbers. All vectors are initialized with random numbers with an uniform distribution within the search bounds of each parameter.

There are several version of DE. Here the rand/1/bin version of DE is used because it is robust and provides the best results for different kind of benchmarks and real optimization problems [10].

Nowadays exits lot of literature about DE, reader can use [11] for a good starting point about deeply DE details.

Algorithm 1. Differential evolution algorithm (rand/1/bin version)

```
Require: The search domain and the value $s$ for the stop condition. The values for
    population size, $\mu$; maximum number of generations, $M$; difference and recombi-
    nation constants, $F$ and $R$ respectively.
Ensure: A solution of the minimization problem
initialize($X = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\}$)
evaluate($X$)
$k = 0$
repeat
    for $j = 1$ to $\mu$ do
        Let $r_1$, $r_2$ and $r_3$ be three random integers in $[1, \mu]$, such that $r_1 \neq r_2 \neq r_3$
        Let $i_{\text{rand}}$ be a random integer in $[1, n]$
        for $i = 1$ to $n$ do
            $x'_{i,j} = \begin{cases} x_{i,r_3} + F(x_{i,r1} - x_{i,r2}) & \text{if } U(0,1) < R \text{ or } i = i_{rand} \\ x_{i,j} & \text{otherwise} \end{cases}$
        $x'_{n+1,j} = \text{evaluate}(\mathbf{x}'_j)$
        if $x'_{n+1,j} < x_{n+1,j}$ then
            $\mathbf{x}_j = \mathbf{x}'_j$
    min $= x_{n+1,1}$, max $= x_{n+1,1}$
    for $i = 2$ to $n$ do
        if $x_{n+1,i} <$ min then
            min $= x_{n+1,i}$
        if $x_{n+1,i} >$ max then
            max $= x_{n+1,i}$
    $k \leftarrow k + 1$
until (max $-$ min) $< s$ or $k > M$
```

The pseudocode of DE is shown in Algorithm 1. The core of DE is in the loop on lines 8-12: a new individual is generated from three different individuals chosen randomly; each value of the new vector (it represents a new individual) is calculated from the first father, plus the difference of the other two fathers multiplied by F, the difference constant; the new vector value is calculated if a random real number (between zero and one) is less than R, the DE's recombination constant. To prevent the case when the new individual is equal to the first father, at least one vector's component is forced to be calculated from their fathers values, it is in line 9 of the pseudocode, when $i = i_{\text{rand}}$, and i_{rand} is a integer random number between 1 and n. Then the new individual is evaluated, if it is better than the father (in lines 11-12), then the child replaces its father. The stop condition used here is: if the number of iterations is greater than 10,000, or when the difference in the objective function values of the worst and best individuals is less than 0.001. This stop condition is called *diff* criteria in [12], and is the recommended for a global optimization task.

According to the test in CEC 2005 conference [13], DE is the second best heuristic to solve real parameter optimization problems, when the number of parameters is around 10. The best heuristic is a Evolution Strategy called G-CMA-ES [14]. DE was chosen because it has a better execution time and it is very easy to implement

4 Experiments and Results with Real Data

Three experiments are carry on public available dataset of Zhang [15]. It consist of five images of a planar calibration pattern of 256 corners, taken with a CCD camera (resolution of 640 × 480). For each image, the corners positions found by Zhang are available. This data set is rather challenging for plane-based self-calibration. Indeed, there are few images (five), small rotations around optical axis, the plane orientation does not vary much, and a significant lens distortion is present.

Fist experiment, is performed with Zhang's images with lens distortion corrected. The camera parameters given in Zhang's paper [4] are $f = 832.53$, $(u_0, v_0) = (303.96, 206.59)$ and $k_1 = -0.288$, $k_2 = 0.190$ for the fist two terms of lens radial distortion. With this information a new set of five corners positions, but now without lens distortion, were generated (considering that the aspect ratio is equal to 1.0).

The search bounds for DE are $-90° \leq \theta_1, \theta_2 \leq 90°$, $-180° \leq \theta_3 \leq 180°$, $-50 \leq t_1, t_2 \leq 50, 100 \leq t_3 \leq 1000, 100 \leq f \leq 2000$, (u_0, v_0) is fixed to $(320, 240)$ (the image center). Here the convention $R = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ is used.

From the five sets of corners points, each one corresponding to one image, there exists 10 combinations of three images. The mean and standard deviation for f for each three images calculated with the proposed algorithm is shown in Table 1. The shown statistics are for 40 executions. On each execution seven runs of the algorithm is made and the median, according the reprojection error, is taken. For this problem a population of 50 individuals, difference constant equal to 0.7, and recombination constant of 0.9 are used.

Table 1. Results for the first experiment with real data. s.d. is standard deviation, r.e. reprojection error. Numbers on first column are three indexes corresponding to each image.

Triplet	f	f s.d.	r.e.	r.e. s.d.
123	816.63	0.38	0.0623	7×10^{-6}
124	836.94	9.90	0.0895	0.13
125	842.09	0.24	0.0262	9×10^{-6}
134	820.86	0.34	0.0451	6×10^{-6}
135	821.56	0.28	0.0491	7×10^{-6}
145	847.04	1.00	0.0175	5×10^{-6}
234	828.71	0.18	0.0681	1×10^{-5}
235	829.78	0.24	0.0745	1×10^{-5}
245	848.83	1.10	0.0173	1×10^{-5}
345	833.40	0.45	0.0552	7×10^{-6}
Total	832.58	11.06	0.0505	0.0478

Table 2. Results for the second experiment. The third column is the value of f obtained with the bundle adjustment of all the five images.

Triplet	Initial f	Final f	% error
123	816.57	828.03	-0.54
124	842.01	826.73	-0.70
125	842.14	826.96	-0.67
134	820.04	828.11	-0.53
135	821.68	828.69	-0.46
145	847.19	827.27	-0.63
234	828.31	827.18	-0.64
235	829.97	827.38	-0.62
245	848.38	827.80	-0.57
345	832.91	827.57	-0.60

The second experiments is using the five images. From every triplet of images in Tab. 1, the other two images are added minimizing (4) for $i = 4, 5$ also with DE. The orientation and position of these two new images are calculated fixing both f and the reconstructed plane ($\hat{\mathbf{P}}_j$ in (4)). To solve this problem the same conditions are used except that the number of individuals is set to 30. Now all the parameters are refined using Bundle Adjustment: this is, (4) is minimized with $i = 1, 2, \ldots, 5$ with the Levenberg-Marquardt method provided by MINPACK (`lmdif1` function) as Zhang suggested in his paper [4]. the initial solution obtained with DE is a local optimum. In order to obtain the global optimum, t_1^i and t_2^i, $i = 1, 2, \ldots, 5$, with values less than 10.0 are set to 0.0 and the bundle adjustment (five steps, or when RMS reprojection error is less than 0.075) is executed. The result is shown in Table 2. The error in the estimation of the focal length is less than 0.7 %. All the bias in error are negatives, and this could be due the skew was made equal to 0.0 instead its real value in this experiment.

The third experiment performs the same calculations that the second experiment but using the raw original data and starting with the principal point fixed at the images center. In the bundle adjustment, the correction of the lens distortion is added, and the principal point values are also refined. Results and errors vs the values obtained by Zhang are shown in Tab. 3.

The camera focal length is estimated with an error less than 1.3% from the results of the third experiment. But the principal point position has an error less than 8%, and error is even more for the lens distortion parameters. According to the simulation results in [6], the focal length is not affected by the principal point position, but the orientation and positions of the involved plane views are affected. Therefore, from this result, it can be seen that the principal point position can not be estimated with accuracy if a reference plane model –which is a calibration method– is not used.

Table 3. Results of the third experiment. Initial values for the principal point are $(u_0, v_0) = (320, 240)$, and distortion parameters $k_1 = 0$, $k_2 = 0$. Final values were obtained with all five images.

Triplet	Initial	Final values					Percentage in error				
	value f	f	u_0	v_0	k_1	k_2	f	u_0	v_0	k_1	k_2
123	894.96	843.21	294.07	208.58	-0.2145	0.2704	1.28	-3.25	0.96	-5.93	42.34
124	814.28	837.39	316.71	206.26	-0.2154	0.1865	0.58	4.19	-0.16	-5.51	-1.86
125	824.02	841.15	294.75	196.13	-0.1863	0.1926	1.04	-3.03	-5.07	-18.30	1.35
134	878.45	838.20	293.54	195.75	-0.2117	0.2671	0.68	-3.43	-5.25	-7.17	40.60
135	879.89	834.27	292.22	195.65	-0.2183	0.2620	0.21	-3.86	-5.29	-4.27	37.88
145	854.62	836.63	295.17	196.71	-0.2208	0.3021	0.49	-2.89	-4.78	-3.17	58.98
234	854.62	840.37	293.87	192.25	-0.2039	0.2410	0.94	-3.32	-6.94	-10.59	26.84
235	857.77	833.33	293.66	190.93	-0.2167	0.2559	0.10	-3.39	-7.58	-4.95	34.68
245	803.48	836.32	294.60	196.86	-0.2217	0.3003	0.46	-3.08	-4.71	-2.76	58.04
345	832.75	833.38	291.79	193.28	-0.2171	0.2546	0.10	-4.00	-6.45	-4.79	34.02

5 Discussion

The convergence of a heuristic is at most linear. Convergence of algorithms based on derivatives is quadratic. Other heuristics could be used to solve non-linear optimization problems, but their theoretical convergence limit is always linear. This argumentation lead to the idea that an heuristic should not be used instead a conventional numerical algorithm. In the methodology proposed in this work, the heuristic differential evolution solves the difficult task of finding an initial reconstructed plane, from three images; and then this reconstructed plane is used to add more images, and perhaps, to correct the camera lens distortion. For these last two tasks, conventional bundle adjustment was used.

From the result of the third experiment is not clear that the proposed method can be used to correct the lens distortion. But the method can be applied on images taken with a modern digital camera where distortion can be ignored [6].

The proposed method solves directly a non-linear optimization problem, and its performance is good under noisy conditions (1-2 pixels) in points positions.

A run of the DE algorithm with the same conditions used in experiments (50 individuals, $s = 0.001$) takes approximately 18 sec [1].

6 Conclusions

A method for direct (without using derivatives) self-calibration of a camera from three images of an unstructured plane was presented. This method uses directly the pinhole camera model to estimate the positions, the orientation and the camera parameters of the three views, and also obtains the reconstructed plane. The reprojection error is minimized. This non-linear optimization problem was solved using the heuristic Differential Evolution. Moreover, more images can

[1] Time was measured in a 1.8 GHz PowerPC G5 Mac with Mac OSX and gcc compiler.

be incorporated, using the reconstructed plane as a model plane. The whole structure then is refined with a conventional bundle adjustment.

The solution with a heuristic has a cost: running time is high compared with a conventional algorithm such as Levenberg-Marquardt, but the advantage is that a starting solution, near to the optimal solution, is not necessary to solve the non-linear problem.

Acknowledgments. This work was partially supported by CONACyT, México under grant 80965.

References

1. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge Uni. Press, Cambridge (2003)
2. Heyden, A., Astrom, K.: Flexible calibration: minimal cases for auto-calibration. In: ICCV-1999, pp. 350–355. IEEE Press, Los Alamitos (1999)
3. Hemayed, E.E.: A survey of camera self-calibration. In: AVSS 2003: Proc. IEEE Conf. on Advanced and Signal Based Surveillance. IEEE Press, Los Alamitos (2003)
4. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. on Patt. Anal. & Mach. Intel. 22, 1330–1334 (2000)
5. Sturm, P.F., Maybank, S.J.: On plane-based camera calibration: A general algorithm, singularities, applications. In: CVPR, pp. 432–437 (1999)
6. Menudet, J.F., Becker, J.M., Fournel, T., Mennessier, C.: Plane-based camera self-calibration by metric rectification of images. Image and Vision Computing 26, 913–934 (2008)
7. Bocquillon, B., Gurdjos, P., Crouzil, A.: Towards a guaranteed solution to plane-based self-calibration. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 11–20. Springer, Heidelberg (2006)
8. Storn, R., Price, K.V.: Differential evolution – a simple and efficient heuristic for global optimization over continuos spaces. J. of Global Optimization 11(4), 341–359 (1997)
9. Herman, G.T.: Image Reconstruction from Projections: The Fundamentals of Computerized Tomography. Academic Press, London (1980)
10. Mezura, E., Velázquez, J., Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: GECCO 2006, New York, NY, USA, pp. 485–492. ACM Press, New York (2006)
11. Chakraborty, U.K.: Advances in Differential Evolution. Studies in Computational Intelligence. Springer, Heidelberg (2008)
12. Zielinski, K., Laur, R.: Stopping criteria for differential evolution in constrained single-objective optimization. In: Advances in Differential Evolution. Springer, Heidelberg (2008)
13. Hansen, N.: Comparisons results among the accepted papers to the special session on real-parameter optimization at CEC 2005 (2006), http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/compareresults.pdf
14. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proceedings of the Congress on Evolutionary Computation CEC-2005, pp. 1769–1776 (2005)
15. Zhang, Z.: http://research.microsoft.com/~zhang/calib/

Graph-Cut versus Belief-Propagation Stereo on Real-World Images

Sandino Morales[1], Joachim Penc[2], Tobi Vaudrey[1], and Reinhard Klette[1]

[1] The *.enpeda..* Project, The University of Auckland, New Zealand
[2] Informatics Institute, Goethe University, Frankfurt, Germany

Abstract. This paper deals with a comparison between the performance of graph cuts and belief propagation stereo matching algorithms over long real-world and synthetics sequences. The results following different preprocessing steps as well as the running times are investigated. The usage of long stereo sequences allows us to better understand the behavior of the algorithms and the preprocessing methods, as well as to have a more realistic evaluation of the algorithms in the context of a vision-based Driver Assistance System (DAS).

1 Introduction

Stereo algorithms aim to reconstruct 3D information out of (at least) a pair of 2D images. To achieve this, corresponding pixels in the different views have to be matched to estimate the disparity between them. There exist many approaches to solve this matching problem, most of them too slow and/or inaccurate. In this paper we compare a graph cut method – which produces in [1,8,13] very good results but is quite slow – and belief propagation stereo which has proven in [5,13] to produce good results in reasonable running time. Both algorithms apply *global* 2D optimization by using information from potentially unbounded 2D neighborhoods for pixel matching, as opposed to, for example, *local* techniques (e.g., correlation-based), or *semi-global* scan-line optimization techniques (e.g., dynamic programming, semi-global matching). Furthermore, we are interested in analyzing various preprocessing methods (as suggested in [5,17]) in order to minimize common issues of real-world imaginary. We are in particular interested in eliminating a negative influence of brightness artifacts, which cause major issues for matching algorithms. This effect on stereo reconstruction quality is often neglected when looking at indoor scenes, with good lighting and cameras. As stated in [9], this kind of noise has a significant influence on the output of stereo algorithms. Following [5], we use the simple Sobel edge detector in order to improve the outcome of the algorithms. We also use *residual images* (i.e., images resulting from subtracting a smoothed version from an original image) that have proved to be of use for overcoming brightness issues, see [17]. The processing time of the algorithms it is also investigated, as this is of importance for most applications, such as vision-based driver assistance systems (DAS) and mobile robotics.

Performances of these two algorithms (BP and GC) have been compared in the past, but only for engineered or synthetic images; see, for example, [15]. Our study is focused on a comparison of the performance of both algorithms on long real-world

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 732–740, 2009.

image sequences; but we also investigate the performance of the algorithms over a long synthetic sequence (for behavior with respect to some systematic changes in this synthetic sequence, but not for ranking of methods; indoor or synthetic data do have limited relevance for the actual ranking of methods for real-world DAS). Those long test sequences allow us to better understand the behavior of the algorithms in general, and in particular the effects of previously proposed preprocessing methods. To overcome the lack of ground truth we use a sequence recorded with three calibrated cameras; thus we are able to use *prediction error analysis* as a quality measure [14]; we use the same approach to evaluate the performance of the algorithms on the chosen long synthetic sequence.

This paper is structured as follows: Section 2 specifies the implementations used in this paper of the graph cut and belief propagation algorithms; it also recalls prediction error analysis and informs about the chosen preprocessing methods. Section 3 presents and discusses the obtained results. Conclusions are stated in Section 4.

2 Approach for Evaluation

The experiments have been performed using a very recent graph cut implementation from V. Kolmogorov and R. Zabih[1] which can detect occlusions quite well; and a modified coarse-to-fine belief propagation algorithm of Felzenszwalb and Huttenlocher[2] as implemented for [5], focusing on more reliable (and time-efficient) matching, therefore using max-product, 4-adjacency, truncated quadratic cost function, red-black speed-up, and coarse-to-fine processing. Both algorithms were implemented under a C++ platform. For a detail discussion on belief propagation and graph cut algorithms see [7] and [3], respectively.

The outline of our experiments is as follows. We evaluate both algorithms over a synthetic and a real-world sequence, and compare results and computational time. Furthermore, we use two different preprocessing methods in order to improve the results. For the graph cut algorithm we also analyze the effect of different number of iterations (between 1 and 3). The algorithms were tested on an Intel Core2 vPro at 3.0 GHz with 4 GB memory using Windows Vista as the operating system.

Data Set. The *POV-ray* synthetic sequence (100 stereo pairs) with available ground truth is from Set 2 on [2], as introduced in [16]. The real-world sequence of 123 frames (recorded with three calibrated cameras using the research vehicle of the *.enpeda..* project, which is the *ego-vehicle* in our experiments) is from Set 5 on [2], as introduced in [10], and it is a fairly representative example of a daylight (no rain) outdoor sequence, containing reflections and large differences in brightness between subsequent stereo pairs, or between the left and right image of the stereo pair. The use of long sequences facilitates the recognition of circumstances that may affect the performance of an algorithm, as well as it helps to understand the *robustness* of an algorithms with

[1] See http://www.adastral.ucl.ac.uk/vladkolm/software/match-v3.3.src.tar.gz

[2] See http://people.cs.uchicago.edu/~pff/bp for original sources.

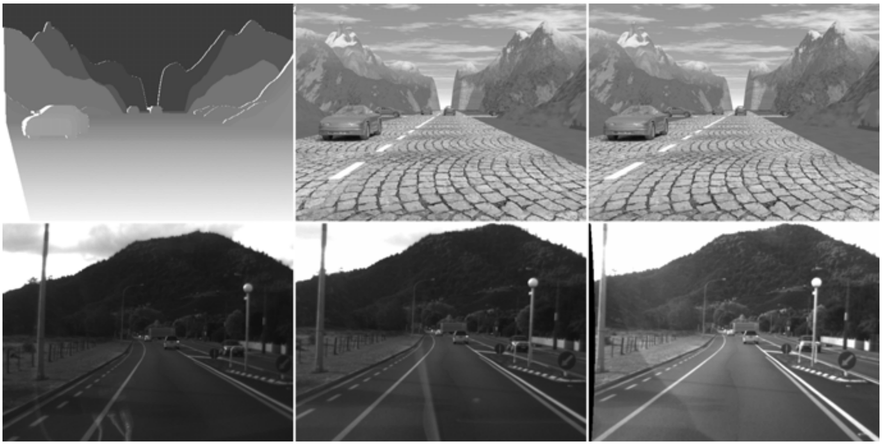


Fig. 1. Data sets. Synthetic sequence (upper row), form left to right: Ground truth disparity (dark = far, light = close, white = occlusion), left and right views of frame 43. Real-world sequence (lower row): from left to right, view of the left, center, and right cameras of frame 37.

respect to changes in circumstances (e.g., brightness differences, lighting artifacts, close objects, night, or rain). See Figure 1 for examples of the used data sets.

Preprocessing Methods. A vision-based DAS has to deal with input data that are recorded under uncontrolled environments. Among all the adverse conditions faced by outdoor image grabbing, brightness differences between the images of a stereo pair have a particularly negative influence on the output of the stereo algorithms [9]. In order to over come this almost unavoidable issue, following [5], we preprocess our sequences using a 3×3 Sobel edge operator (with a processing time of 0·06 s per stereo pair) to create an *edge sequence*. In [12], the simple Sobel operator proved to be the most effective edge-operator within a group of edge operators, tested for improving correspondence analysis on real-world data.

In [17], the authors used *residual images* to remove the illumination differences between correspondence images. We analyze whether there is an improvement in the output of the belief propagation and graph cut stereo algorithms using residual images as source data. Given an image I, we consider it as a composition $I(p) = s(p) + r(p)$, for pixel position $p \in \Omega$ (the image domain), where $s = S(I)$ denotes the *smooth component* and $r = I - s$ the *residual* one. We use the straightforward iteration scheme to obtain the residual component of the image I:

$$\mathbf{s}^{(0)} = I, \quad \mathbf{s}^{(n+1)} = S(\mathbf{s}^{(n)}), \quad \mathbf{r}^{(n+1)} = I - \mathbf{s}^{(n+1)}, \quad \text{for } n \geq 0.$$

In our experiments we use a 3×3 mean filter to generate the smooth component and $n = 40$ iterations (with a computational time of 0·07 s per stereo pair; see [17] for a reasoning for selecting mean filtering and $n = 40$). We refer to the sequence formed by residual images as *residual sequence*. See Figure 2 for a sample stereo pair of the real-world residual sequence and the edge synthetic sequence.

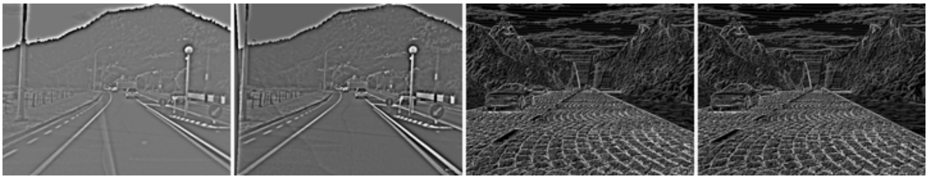


Fig. 2. Examples of the output of the preprocessing methods. Left: Residual stereo pair frame 37 of the real-world sequence. Right: Sobel edge stereo pair frame 43 of the synthetic sequence.

Evaluation approach. To objectively evaluate the performance of the algorithms over the real-world sequence (with non-available ground truth), the output of the algorithms is analyzed using the so-called *prediction error* [14]. This technique requires at least three images of the same scene: two of them are used to calculate a disparity map, while the third one is used for evaluation purposes. For consistency, the evaluation of the synthetic sequence is also performed with the prediction error. The third or virtual image used to evaluate the results is generated using the same pose of the left-most camera of the three-camera set-up in our research vehicle (while recording the real-world sequences) and the available ground truth.

We follow the method described in [10], where the (rectified) images recorded by the center and right-most camera are used as the input data of the stereo algorithms. The resultant disparity map and the center image are used to generate (by geometrical means) a virtual image as it would be recorded by the left-most camera. This virtual image is then compared with the actual left-most image in the following way: for each frame t of the given trinocular sequence, let Ω_t be the set of all pixels in the left image I_l, such that their source scene point is also visible in the center and right images. Let (x, y) be the coordinates of a pixel in Ω_t with intensity $I_l(x, y)$. The method above assigns to the pixel with coordinates (x, y) in the virtual image I_v an intensity value $I_v(x, y)$ (defined by the intensity of a certain pixel in the center image). Thus, we are able to compute the *root mean squared* (RMS) error between the virtual and the left image as follows:

$$R(t) = \frac{1}{|\Omega_t|}\Big(\sum_{(x,y)\in\Omega_t} [I_l(x, y) - I_v(x, y)]^2\Big)^{1/2}$$

where $|\Omega_t|$ denotes the cardinality of Ω_t. A *high* RMS means there is more error.

The *normalized cross correlation* (NCC) is also used to compare left and virtual image, applying the following:

$$N(t) = \frac{1}{|\Omega_t|} \sum_{(x,y)\in\Omega_t} \frac{[I_l(x, y) - \mu_r][I_v(x, y) - \mu_v]}{\sigma_r \sigma_r}$$

μ_l and μ_v denote the means, and σ_l and σ_v the standard deviations of I_l and I_v, respectively. A *low* NCC means there is more error.

Since we are dealing with a image sequence, the results can be graphed over time (see Figure 5 for an example). To summaries the large dataset, we compute the mean and the zero mean standard deviation (ZMSD - the standard deviation assuming a mean of zero) of the results in a sequence.

3 Results and Discussion

Synthetic Sequence. According to [1], GC only needs a few iterations to obtain acceptable results. Thus we test the algorithm with only one or three iterations over the three sequences. For all the sequences, differences in results for either one or three iterations are almost imperceivable, visually and statistically. The RMS metric reports a slight improvement using three iterations, and the NCC shows that the results are a bit better using just one (see Table 1). The computational time, on average, was 135·3 s and 386·7 s for one and three iterations, respectively. The latter result discourages the use of more than one iteration for this synthetic sequence.

Differences in GC results between the preprocessed sequences and the original ones are not consistent either. On one hand, NCC reports that the best performance is with the original sequence. On the other hand, the best RMS results are obtained with the edge sequence. Visually, NCC seems to report more accurately the behavior of the algorithm, as the results seems to suffer degradation with the preprocessed sequences (see Figure 3).

BP shows a different behavior. With RMS, the best overall results were obtained with the original sequence, while with NCC this sequence showed the worst performance. Again, by visual inspection, NCC seems to reflect better the performance of the algorithms, as the results get better (visually) using any of the discussed preprocessing methods. The average computational time was 98·5 s (parameters used: ITER $= 7$, LEVELS $= 6$, DISC$_K = 50$, DATA$_K = 35$, $\lambda = 0{\cdot}07$).

Summarizing (see BP values in Table 1), the metrics show contradictory results when using preprocessing methods. Visually, NCC seems to be the more appropriate metric; following the NCC results, GC has a better performance than BP on the original synthetic sequence, with one or three iterations; but, with preprocessing, BP produces results are as good as GC. See Figure 3.

Real-world Sequence. With GC, all the sequences reported worse RMS results with three iterations compared to one! NCC reports basically no change except for the edge sequence, for which the results are slightly worse for three iterations (see Table 2).

Table 1. Summarizing NCC and RMS results for the synthetic sequence

Evaluation		GC - 1 Iteration		GC - 3 Iterations		BP	
Approach	Sequence	Mean	ZMSD	Mean	ZMSD	Mean	ZMSD
NCC	Original	0·76	0·76	0·75	0·75	0·68	0·68
	Residual	0·75	0·75	0·74	0·74	0·76	0·76
	Sobel	0·73	0·73	0·72	0·72	0·73	0·73
RMS	Sobel	36·04	36·07	36·02	36·04	36·02	36·04
	Residual	36·68	36·70	36·65	36·67	36·51	36·54
	Original	36·82	36·84	36·66	36·68	35·86	35·88

Table 2. Summarizing NCC and RMS results for the real world sequence

Evaluation Approach	Sequence	GC - 1 Iteration Mean	 ZMSD	GC - 3 Iterations Mean	 ZMSD	BP Mean	 ZMSD
NCC	Residual	0·66	0·67	0·66	0·67	0·68	0·69
	Sobel	0·66	0·67	0·65	0·65	0·65	0·66
	Original	0·64	0·65	0·64	0·65	0·65	0·66
RMS	Residual	33·48	34·06	33·50	34·08	32·91	33·48
	Sobel	34·03	34·62	34·19	34·78	33·00	33·58
	Original	35·34	35·94	35·49	36·10	34·04	34·57

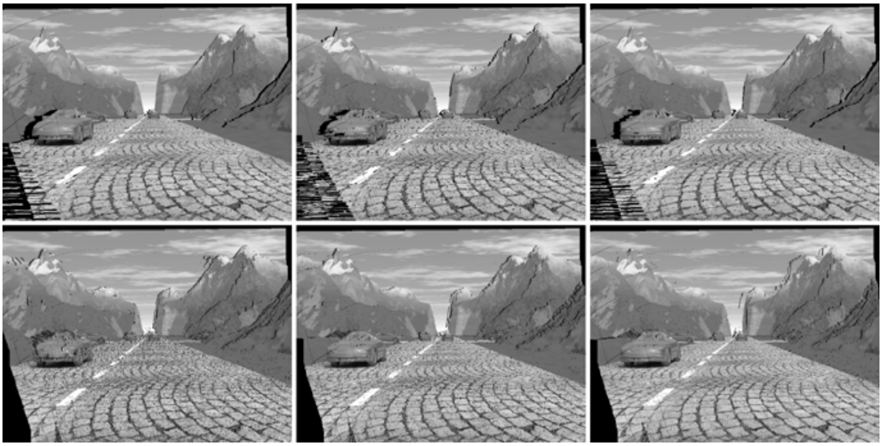


Fig. 3. Examples of the generated virtual view. Left to right: Original, edge map and residual sequences. Upper row: GC with one iteration. Lower row: BP.

In contrast, the preprocessing methods have a positive influence on the outcome of GC, the residual sequence having the best performance (no matter whether one or three iterations, and for both metrics). The mean computational time was 178·64 s and 390·72s for one and three iterations, respectively, per stereo pair. This result discourages the use of more than one iteration.

BP shows here a similar behavior as GC: the results improved with both preprocessing methods (with respect to both metrics), the residual sequence having the best results. The average NCC does not report any improvement with the edge sequence (when comparing with the original sequence); however, visually, the improvement is obvious when the difference in brightness is large between both images in an input stereo. The parameters used with the real world sequence were as follows: LEVELS $=$ 6, DISC$_K$ $= 500$, DATA$_K$ $= 100$, $\lambda = 0.3$, with an average computation time of 122·44 s per stereo pair of images.

Table 2 illustrates that BP outperforms GC in the overall results (as well as in computational time) even when comparing the best GC result (one iteration over the residual sequence) and the worst of BP (original sequence). Both algorithms

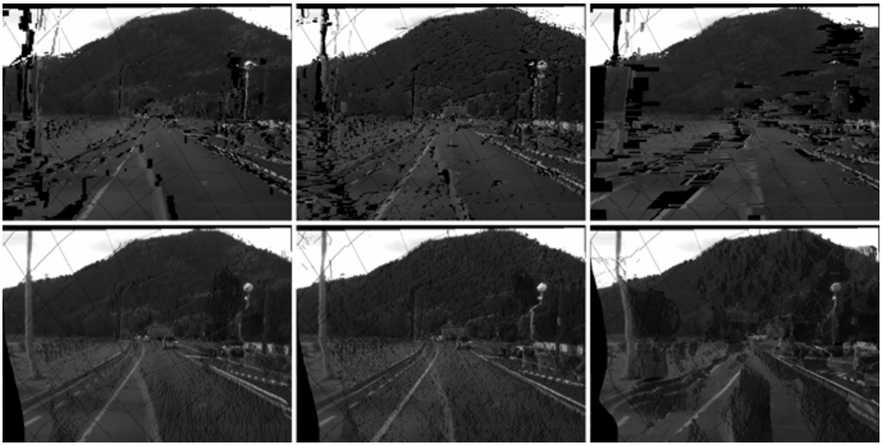


Fig. 4. Examples of the generated virtual view. Left to right: Residual, edge map and original sequences. Upper row: GC with 1 iteration. Lower row: BP.

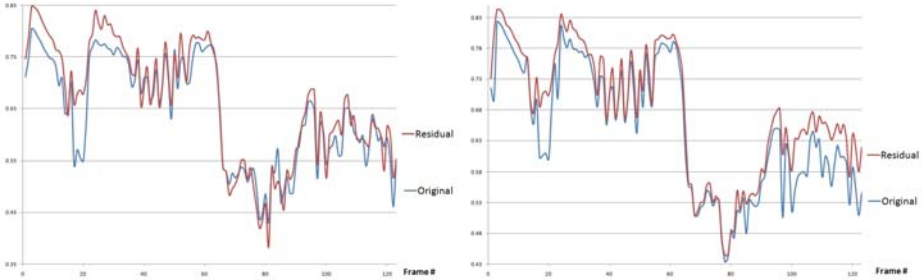


Fig. 5. NCC evaluation for the real-world sequence. Comparison shown between the original and the residual sequence. Left: GC with one iteration. Right: BP.

improve their results using preprocessed sequences, particularly, when the differences in brightness between both images in a stereo input pair are large. Figure 4 shows the calculated virtual views for frame #47 for the original (right) and the preprocessed sequences (left and center); the improvement can be detected visually. It is also interesting to note that there is a minimal (or no) improvement with the preprocessing of the original sequence when differences in brightness are only minor, meaning, that with fairly good balanced images, there would be no need of preprocessing.

Summary. The difference between the computational time between one and three iterations of GC, and the almost null benefit (or even a degradation in the obtained) results, discourages the use of more than one iteration, for both the real-world and the synthetic sequences (and its respective modifications). The preprocessing methods reported better results for both algorithms (except for GC when the synthetic sequence was evaluated), the residual image method having the best performance, see Figure 5. From Table 1, GC outperforms BP over the original synthetic sequence. However, BP had a better performance over the original real-world sequence, showing that it is misleading to evaluate over synthetic sequences when ranking stereo algorithms. This also tells us that more research needs to be done for studying the performance of stereo algorithms different circumstances (night, rain, etc.). For example, in a more recent comparison, GC has shown a better performance on sequences captured in the night, or when objects appear close to the ego-vehicle.

Note that the metrics reported different rankings when evaluating the synthetic sequence, NCC being the one that seems to confirm what can be concluded by visual inspection. This behavior (inconsistency in metrics) was not expected in images that have been recorded under perfect conditions. However, RMS is certainly a 'very accurate' measure, 'asking for to much', and seems to be misleading in evaluations.

4 Conclusions

In this paper we compare the performance of a belief propagation stereo algorithm with a graph cut stereo implementation, using two long sequences (real-world and synthetic) and two different preprocessing methods. We also tested the influence of the number of iterations for the GC algorithm. The different rankings obtained by the algorithms on either the real-world or the synthetic sequence support the usage of a wide class of data sets for testing the performance of the algorithms, to avoid some bias. The preprocessing methods proved to be a good option when dealing with real world images, as the results improved for both algorithms. For the synthetic sequence the metrics do not show consistent results, and when some improvement was detected, then it was only fairly minor. This is as expected, as there is no need to improve 'perfect' (good contrast and grey value distribution) images. We also noticed that there is no need to use more than one iteration with GC; if there was an improvement, it was almost imperceivable, statistically and visually. On the other hand, the difference in computational time is considerably large.

Future work may include the investigation of more metrics and preprocessing methods, as well as the usage of data sets with other adverse conditions such as rain, night time, and so forth.

References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Analysis Machine Intelligence 23, 1222–1239 (2001)
2. .enpeda.. image sequence analysis test site (EISATS), `http://www.mi.auckland.ac.nz/EISATS/`
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Computer Vision 70, 41–54 (2006)
4. Guan, S., Klette, R.: Belief-propagation on edge images for stereo analysis of image sequences. In: Sommer, G., Klette, R. (eds.) RobVis 2008. LNCS, vol. 4931, pp. 291–302. Springer, Heidelberg (2008)
5. Guan, S., Klette, R., Woo, Y.W.: Belief propagation for stereo analysis of night-vision sequences. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 932–943. Springer, Heidelberg (2009)

6. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
7. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Analysis Machine Intelligence 26, 65–81 (2004)
8. Kolmogorov, V., Zabih, R.: Graph cut algorithms for binocular stereo with occlusions. In: Paragios, N., Chen, Y., Faugeras, O. (eds.) Handbook of Mathematical Models in Computer Vision, pp. 423–438 (2006)
9. Morales, S., Vaudrey, T., Klette, R.: An in depth robustness evaluation of stereo algorithms on long stereo sequences. In: Proc. IEEE Intelligent Vehicles Symp., pp. 347–352 (2009)
10. Morales, S., Klette, R.: A Third Eye for Performance Evaluation is Stereo Sequence Analysis. In:Proc. CAIP (to appear, 2009)
11. Ohta, Y., Kanade, T.: Stereo by intra- and inter-scanline search using dynamic programming. IEEE Trans. Pattern Analysis Machine Intelligence 7, 139–154 (1985)
12. Al-Sarraf, A., Vaudrey, T., Klette, R., Woo, Y.W.: An approach for evaluating robustness of edge operators on real-world driving scenes. In: IEEE Conf. Proc. IVCNZ 2008, Digital Object Identifier 10.1109/IVCNZ.2008.4762096 (2008)
13. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision 47, 7–42 (2002)
14. Szeliski, R.: Prediction error as a quality metric for motion and stereo. In: Proc. Int. Conf. Computer Vision, vol. 2, pp. 781–788 (1999)
15. Tappen, M., Freeman, W.: Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In: Proc.9th IEEE ICCV, vol. 2, pp. 900–906 (2003)
16. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behavior on synthetic and real-world stereo sequences. In: Proc. Int. Conf. Image Vision Computing, New Zealand. IEEE Xplore, Los Alamitos (2008)
17. Vaudrey, T., Klette, R.: Residual images remove illumination artifacts for correspondence algorithms!. In: Proc. Pattern Recognition - DAGM (to appear, 2009)

Combining Appearance and Range Based Information for Multi-class Generic Object Recognition

Doaa Hegazy and Joachim Denzler

Institute of Computer Science, Friedrich-Schiller-University in Jena,
Ernst-Abbe-Platz 2, 07743 Jena, Germany
doaa.hegazy@uni-jena.de, joachim.denzler@uni-jena.de

Abstract. The use of range images for generic object recognition is not addressed frequently by the computer vision community. This paper presents two main contributions. First, a new object category dataset of 2D and range images of different object classes is presented. Second, a new generic object recognition model from range and 2D images is proposed. The model is able to use either appearance (2D) or range based information or a combination of both of them for multi-class object learning and recognition. The recognition performance of the proposed recognition model is investigated experimentally using the new database and promising results are obtained. Moreover, the best performance gain by combining both appearance and range based information is 35% for single classes while the average gain over classes is 12%.

1 Introduction

Generic object recognition (GOR) has been an important topic of the computer vision research in recent years. Many different approaches have been developed to give a solution to such difficult problem (e.g. [2, 7]). However, most of the successful approaches developed up to date have concentrated on generic recognition of objects from 2D images, and very little attention has been paid to the use of 3D range data. Range images have the advantage of providing direct cues of the shape of objects which is important for representing and recognizing different visual object classes.

However, the absence of GOR work using range images is due to two main reasons: 1) the non-availability of an object category dataset which provides range data (images) of its member classes. The currently available object category datasets which emerged as standards for the GOR community provide only 2D images of their object categories such as Caltech-6, Caltech 101 [1] and Graz [7], 2) surface shape representation is very important in a recognition procedure from range data in general. However, it is not clear which representation is more suitable for learning shapes of visual object classes. Authors in [8] have developed an approach to recognize objects belonging to a particular shape class in range images and presented a shape representation called symbolic surface signature. The dataset used for learning and classifying their model is a set of range

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 741–748, 2009.

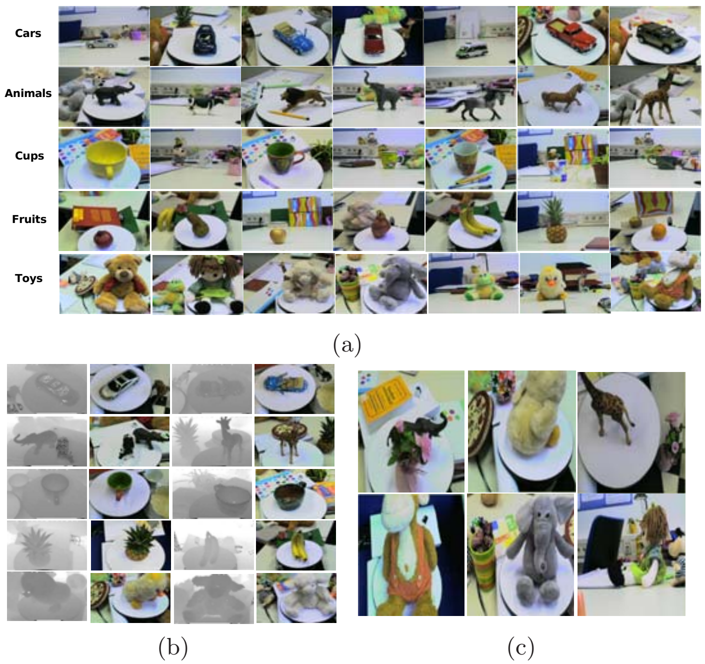
images of objects made of clay. Many important differences exist between the model proposed here and the one in [8]. Among these differences is that in our approach, combination of appearance (2D) and range based information is used for GOR recognition which is not the case in [8].

This paper addresses the use of range images for GOR and presents two main contributions. First, a new object category dataset is constructed. The dataset provides 2D (color) as well as 3D range images of its member classes, with dense background clutter and occlusion. The availability of the two different image types makes the dataset suitable for GOR from either 2D or range data or from a combination of both data types. Moreover, it can be used for both 2D and 3D GOR as well. The second contribution is a new model for GOR with the following advantages: 1) it recognizes generic object classes from range images by exploiting shape cues of objects, 2) it is based on local representation of range images by using interest regions detection and description. Therefore, the model is able to recognize objects in range images despite background clutter and occlusion, 3) performs multi-class learning and recognition of generic object classes from range images, 4) the general framework of the recognition model allows the use of 2D images as well for recognition using either texture or color information or both of them and 5) the framework gives the ability to combine both appearance (2D) and shape (range) cues for GOR of multiple classes.

The outline of the remainder of this paper is as follows. Section 2 is devoted to describe the new object category dataset. The proposed generic object recognition model is described in section 3. Experimental evaluations of the proposed model as well as results obtained are presented in section 4. Conclusions are finally drawn in section 5.

2 An Object Category Dataset

An object category dataset of 4220 2D/3D images (2D colored and range images) of 35 objects was constructed using a 3D Time-of-Flight (TOF) PMD camera [4] and a CCD camera. The objects are instances of five visual classes (categories): cars, toys, cups, fruits and animals (see figure 1 (a)). For each object category, seven individual instances were used. Due to the difficulty to record different outdoor views of natural objects using the TOF camera, indoor views in an office environment were captured. Artificial objects were used in replacement of real instances of some visual classes (namely cars and animals) in building the dataset. The instances of each object class were chosen with different sizes and appearance to achieve large intra-class variabilities as much as possible (see figure 1 (a)). Many images of the dataset contain multiple instances of the same class or from different classes. Moreover, the images contain large viewpoint and orientation variations, partial occlusion (e.g. by other objects) and truncation (e.g. by the image boundaries) as well as background clutter (see figure 1 (c)). The images of each individual object instance were acquired under eight different viewing angles and four different heights. This is accomplished as follows: at each height, each object instance was placed on a turn table which was rotated



(a)

(b)

(c)

Fig. 1. (a) Example images of the five object classes of the new object category dataset. (b) Example range range images (using TOF camera) and their corresponding color images (using CCD camera). TOF cameras produce reflected images with respect to images produced by CCD cameras. (c) Example images of the dataset with occlusion and truncation.

through 360 degrees about the vertical axis and eight colored and range images were acquired; one at every 45 degrees. The total number of images acquired using each camera is 32 images for each object instance (4 heights × 8 angles). The TOF camera delivers also an intensity image corresponding to each range image. However, the delivered intensity image is of low resolution which affects direct application of some image processing algorithms on them.

The dataset will be available for public use[1] as we believe that the dissemination and use of this dataset will allow realistic comparative studies as well as a source to test data for development for new techniques of GOR from range images.

3 The Recognition Model

Figure 2 displays a semantic view of the general framework of the proposed GOR model. As shown in the figure, the recognition model consists of three main steps, which are described in this section.

[1] http://www.inf-cv.uni-jena.de/index.php?id=dataset

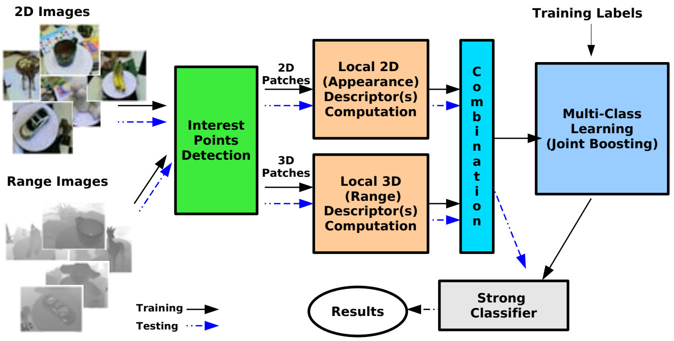

2D Images
Training Labels
Range Images
Interest Points Detection
2D Patches
3D Patches
Local 2D (Appearance) Descriptor(s) Computation
Local 3D (Range) Descriptor(s) Computation
Combination
Multi-Class Learning (Joint Boosting)
Strong Classifier
Results
Training
Testing

Fig. 2. The general framework of the proposed GOR model

3.1 Interest Points Detection

An interest point detector, namely the Hessian affine-invariant point detector [6], is run on 2D images to detect a set of interest points. For the range images, the interest point detector is run on the intensity images corresponding to range images delivered by the TOF camera. Afterwards, the 3D regions corresponding to the detected points are extraced. However, the range data of a TOF camera suffer from large amount of noise. In order to filter some of this noise and smooth the range data, a preprocessing step by applying a median filter is first performed. Furthermore, an initial histogram normalization is applied to the TOF intensity images to enhance their low contrast before interest points detection.

3.2 Local Description

Based on the type of the extracted region (2D or 3D), a local descriptor is computed from it. A set of different local descriptors is used for both types of data including grayscale, color and shape descriptors.

2D Descriptors. Two different types of descriptors are used: the SIFT descriptors [5] and the opponent color angle descriptors [10].

3D (Range) Descriptors. Three shape-specific local feature histograms are used. These features were presented and used in [3] for the task of free-form specific 3D object recognition. The features are namely: pixel depth, surface normals and curvature. The main advantages of these features are that they are easy to calculate, robust to viewpoint changes and contain discriminative information [3]. For the lack of available space, more information about the features are found in [3].

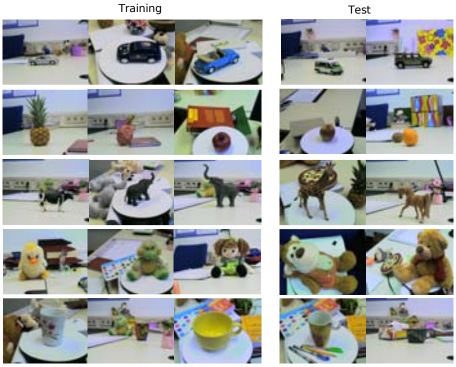


Fig. 3. Object class instances used to train and test the proposed model

3.3 Learning Model

Learning in our model is based on the Joint Boosting algorithm [9] which depends on training multiple binary classifiers at the same time and sharing features among them. The algorithm has the advantage that less training data is needed since many classes can share similar features. Readers are invited to consult [9] for details about the algorithm. In contrast to [9], in our model, combined features are shared among the classes instead of sharing a single feature. This is done through the weak learner used by our learning model (presented in [7]) which is different from the one used in [9].

4 Experimental Evaluations

The presented GOR model is evaluated experimentally to analyze its benefits and limitations. The performance is measured in three cases. First, using only appearance information for recognition. Second, using only 3D (range) information. Finally, using a combination of both types of information. The model, for each previously mentioned case, is trained in two ways: 1) independently, 2) jointly with feature sharing among classes. For all experiments, the number of training iterations (number of weak hypotheses) is fixed to $T = 150$ and is independent of the number of classes. In contrast to the learning model in [9], we are not searching and comparing the learning effort for a certain error rate but we report the ROC-equal-error rate for a certain learning effort, namely T weak hypotheses. All experiments are performed using our new object category dataset. We use five classes: cars, fruits, animals, toys and cups (see figure 3). The number of training examples for each class is 100 examples which results in a total of 500 training examples. For testing, 60 examples per class (images of new instances) are used (a total of 300 examples).

Recognition using appearance information only: The aim of this set of experiments is to measure the performance of the model using only appearance-based

Table 1. Comparison of the ROC-equal-error rates of the appearance (2D) and range based descriptors used separately and combined for learning and classifying the five object classes (with independent learning)

Descriptors	Cars	Fruits	Animals	Toys	Cups	Avg. error over classes
Appearance-Desp.	0.300	0.200	0.250	0.283	0.233	0.253
Range-Desp.	0.417	0.500	0.317	0.183	0.363	0.356
Appearance + range Comb.	0.370	0.167	0.200	0.183	0.183	0.221

Table 2. Confusion matrices of multi-class recognition (with independent learning) results using appearance (2D) descriptors, range descriptors and the appearance-range combination respectively. For the computation of the confusion matrix, the best classification of an image over all classes is counted as the object category. Numbers represent percentage (%) of test images (60 images per class) classified for each class. Columns represent true classes.

	Appearance Desp.					Range Desp.					App.+Range Comb.				
Class	c1	c2	c3	c4	c5	c1	c2	c3	c4	c5	c1	c2	c3	c4	c5
Cars:c1	**47**	1	3	8	3	**70**	**82**	32	25	**65**	**53**	5	5	5	7
Fruits:c2	13	**77**	12	1	12	2	17	1	8	2	15	**80**	8	5	3
Animal:c3	15	2	33	1	5	18	1	**37**	5	18	3	0	**32**	0	7
Toys:c4	7	12	13	**53**	3	5	0	11	**60**	0	12	5	23	**78**	5
Cups:c5	18	8	**38**	27	**77**	5	0	10	2	15	17	10	**32**	12	**78**

information. A combination of the SIFT and color descriptors is used for learning and recognition. Learning the five classes is performed independently. The recognition performance (ROC-equal-error rates) using the test images is displayed in table 1. It should be noted that in our learning model, the background class for learning and classifying one object class (or a subset of classes) is a combination of the other classes. Multi-class learning and classification in such a case is a difficult task as the background class is very heterogeneous in appearance and is much more likely to appear than the various object classes since most of the image is background. This affects in turn the final recognition performance of the model.

Recognition using range information only: A combination of the local shape descriptors (range-based information) is used here alone for learning and recognition. Table 1 displays the recognition performance. Generally, the recognition performance using range-based information is lower than the performance using the appearance-based information. This could be argued to the low resolution of the intensity images of the TOF camera, which are used for point detection when range images are used for recognition. This low resolution of the images (which are, additionally, full of background clutter) affects the detection performance of the point detector and influences, in turn, the classification

Table 3. Comparison of the ROC-equal-error rates of the appearance (2D) and range based descriptors used separately and combined for learning and classifying the five object classes (with joint learning)

Descriptor	Cars	Fruits	Animals	Toys	Cups	Avg. error over classes
Appearance-Desp.	0.267	0.250	0.220	0.350	0.317	0.280
Range-Desp.	0.400	0.550	0.440	0.260	0.350	0.400
Appearance + range Com.	0.383	0.300	0.280	0.217	0.250	0.306

performance. Moreover, the noisy nature of the TOF range images affects the construction of a clear shape representation for each class which has an effect on the recognition performance.

Recognition using a combination of appearance and range based information: To assess the performance of the model when using different types of information (appearance and range), a combination of the appearance and range based information is used for training and testing the recognition model. Again, learning is performed independently. The recognition performance is shown in table 1. The combination of the three different types of descriptors improves the performance over almost all classes. The performance gain using the appearance-range combination is 35% for the best single class (toys) and 12% over classes, which reveals the benefits of combining both different types of information for recognition. Table 2 shows the confusion matrices of recognition using appearance-based information, range-based information and appearance-range combination respectively. The confusion using only range information is high in comparison to the case of appearance information, while the confusion is improved by using appearance-range combination.

Joint vs. Independent Learning: To assess the performance of joint learning, the experiments are repeated with the classes being learnt jointly with feature sharing. Table 3 displays the results of joint learning. It can be noted that the joint learning (table 3) is not suitable for our recognition case as it does not significantly achieve better performance than the independent learning (table 1). The recognition performance using idependent learning is better (achieves lower error rates) in most of the cases than joint learning.

5 Conclusions

This paper has presented two contributions. First, a new object category dataset has been constructed. The dataset has the advantage over existing datasets that it provides both 2D and range data of its member classes and can be used for both 2D and 3D generic object recognition (GOR). Second, a GOR model for multi-classification of visual object classes from range images using shape information has been proposed. Also, the general framework of the model allows the use of appearance-based information extracted from 2D images for recognition. Moreover, it is able to exploit a combination of both appearance-based (extracted from 2D images) and shape-based (extracted from range images) information for recognition of multiple object classes. Experimental evaluation of the model using the two different types of information has shown good performance. However, combining the two different information types improves the recognition performance.

References

[1] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop, p. 178 (2004)

[2] Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: IEEE Computer Society Conference on computer vision and Pattern Recognition CVPR3, June 2003, vol. 2, pp. 264–271 (2003)

[3] Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3d object recognition from range images using local feature histograms. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2001), vol. 2, pp. 394–399 (2001)

[4] Lange, R.: 3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology, PhD thesis, University of Siegen (2000)

[5] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)

[6] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)

[7] Opelt, A., Pinz, A.: Object localization with boosting and weak supervision for generic object recognition. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 862–871. Springer, Heidelberg (2005)

[8] Ruiz-correa, S., Shapiro, L.G., Meil, M.: A new paradigm for recognizing 3-d object shapes from range data. In: Proceedings of the IEEE Computer Society International Conference on Computer Vision 2003, vol. 2, pp. 1126–1133 (2003)

[9] Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 854–869 (2007)

[10] van de Weijer, J., Schmid, C.: Coloring local feature extraction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 334–348. Springer, Heidelberg (2006)

Dealing with Inaccurate Face Detection for Automatic Gender Recognition with Partially Occluded Faces*

Yasmina Andreu, Pedro García-Sevilla, and Ramón A. Mollineda

Dpto. Lenguajes y Sistemas Informáticos
Universidad Jaume I. Castellón de la Plana, Spain
{yandreu,pgarcia,mollined}@uji.es

Abstract. Gender recognition problem has not been extensively studied in situations where the face cannot be accurately detected and it also can be partially occluded. In this contribution, a comparison of several characterisation methods of the face is presented and they are evaluated in four different experiments that simulate the previous scenario. Two of the characterisation techniques are based on histograms, LBP and local contrast values, and the other one is a new kind of features, called Ranking Labels, that provide spatial information. Experiments have proved Ranking Labels description is the most reliable in inaccurate situations.

1 Introduction

Over the past decades, a great number of papers have been published in the face analysis area. Most of them dealt with face recognition [1,2] and face detection [3,4,5,6] problems. However, automatic gender classification has recently become an important issue in this area. Gender recognition has applications in several fields, such as, in demographic data collection, and it also could be an interesting starting point for other face image processes.

According to recent papers [3,4], face detection tasks obtain quite impressive results, although they do not reach 100% accuracy in all situations. Moreover, faces could be occluded by pieces of cloth, such as, scarves or glasses. Consequently, we will focus on the gender recognition problem when the face is not accurately detected and only a partial view of the face is available.

In this paper, we compare several characterization techniques in order to find out which one performs better with the previous restrictions. All these techniques consider a set of $N \times N$ windows over each face image. A feature vector is extracted from each individual window in order to characterize the face. The techniques used are: a well-know method based on Local Binary Patterns (LBPs) which have achieved good results in the face recognition task [2], a description

* This work was partially funded by Projects Consolider Ingenio 2010 CSD2007-00018, DPI2006-15542, and AYA2008-05965-C04-04 from the Spanish Ministry of Science and Innovation, and P1-1B2007-60 from the Fundació Caixa Castelló-Bancaixa.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 749–757, 2009.

based on Local Contrast Histograms (LCH) which can be used independently or together with the LBP [7] and the features proposed by the authors that have been specifically designed to keep not only the contrast information but also the positional information of each pixel inside its window [8].

The rest of the paper is organized as follows: the face descriptions used are introduced in Section 2; in Section 3, the experimental set-up is described in detail; in Section 4, the results are shown and discussed. Finally, our conclusions are given in Section 5.

2 Face Descriptions

This section presents all the face characterization methods used in the experiments, including our features called Ranking Labels.

All the face descriptions use a window that scans the face image to obtain the feature vectors that will characterize the corresponding face. Two of the characterization methods considered are based on histograms computed over the image window (LBP and LCH) while the other method assigns a label to each pixel in the window in such a way that it keeps the information about the position of the pixels inside it.

2.1 Local Binary Patterns

The LBP operator was originally defined to characterize textures. It uses a binary number (or its equivalent in the decimal system) to characterize each pixel of the image. In the most basic version, to obtain this number, a 3×3 neighborhood around each pixel is considered. Then, all neighbors are given a value 1 if they are brighter than the central pixel or value 0 otherwise. The numbers assigned to each neighbor are read sequentially in the clockwise direction to form the binary number which characterize the central pixel. The texture patch in a window is described by the histogram of the LBP values of all the pixels inside it.

To deal with textures at different scales, the LBP was extended to use neighborhoods of different radii. The local neighborhood is defined as a set of sampling points spaced in a circle centered at the pixel to be labeled. A bilinear interpolation is used when a sample point does not fall in the center of a pixel. In the following, the notation $\text{LBP}_{P,R}$ will be used to refer to LBP that uses a neighborhood with P sample points on a circle of radius R.

The LBP operator can be improved by using the so-called uniform LBP [9]. The uniform patterns have at most two one-to-zero or zero-to-one transitions in the circular binary code. The amount of uniform LBP (LBP^u), when a 8-neighborhood is considered, is 58. However, a histogram of 59 bins is obtained from each window, since the non-uniform patterns are accumulated into a single bin. Although the number of patterns is significantly reduced from 256 to 58; it was observed that the uniform patterns provide the majority of patterns, sometimes over 90%, of texture [10].

The LBP operator gives more significance to some neighbors than to others, which makes the representation sensitive to rotation. In order to obtain a LBP

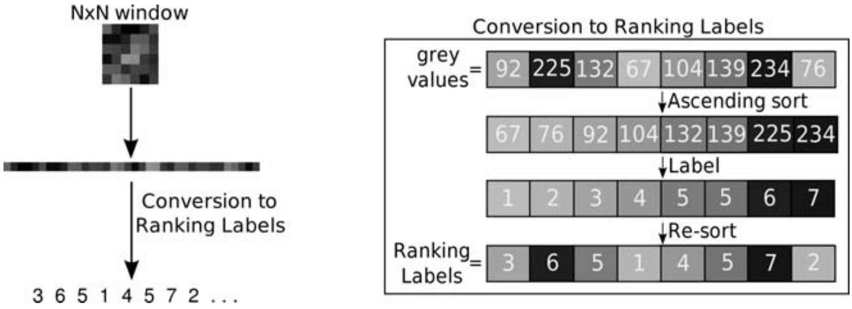

NxN window
Conversion to Ranking Labels
3 6 5 1 4 5 7 2 ...
Conversion to Ranking Labels
grey values = 92 225 132 67 104 139 234 76
Ascending sort
67 76 92 104 132 139 225 234
Label
1 2 3 4 5 5 6 7
Re-sort
Ranking Labels = 3 6 5 1 4 5 7 2

Fig. 1. Example of the extraction process of Ranking Labels

rotationally invariant [9], all possible binary numbers that can be obtained by starting the sequence from all neighbors in turn are considered. Then the smallest of the constructed numbers is chosen. In case the face is slightly inclined in the image, the rotation invariant uniform LBP (LBP^{ri,u}) is supposed to provide a more accurate description of the face. As the quantity of LBP^{ri,u} is 9 in this case, a histogram of 10 bins describes each window.

2.2 Local Contrast Histograms

When computing the LBPs the information about the contrast in the window is lost. Therefore, local contrast histograms (LCH) can be used as an alternative feature set or combined together with LBPs in order to complement their characterization [7].

To compute the local contrast value of a pixel, a neighborhood is defined in a similar way as for LBP. Then the average of the grey level values of those neighbors that are brighter than the central pixel is subtracted from the average of the grey level values of the darker ones. Finally, all the local contrast values are accumulated in a histogram to obtain the LCH$_{P,R}$. This notation means that the neighborhood used has P sample points on a circle of radius R. In order to have the same number of features as for the LBPs, when the neighborhood used has 8 samples points and its radius is 1 the LCH has 10 bins, whereas if the radius is 2 a 59-bin histogram is obtained.

2.3 Ranking Labels

In this description method a vector of ranking labels characterizes each window. For a $N \times N$ window the values of the pixels within the window are substituted by their ranking positions. In other words, the grey level of each pixel is replaced by a numeric label that represents its position in the sorted list in ascending order of all grey levels within the window. This provides a more robust feature vector while keeping the positional information of each pixel inside the corresponding window. This characterization process is shown in Fig. 1 (see [8] for more detail).

3 Experimental Set-Up

3.1 General Methodology

The methodology designed uses the full-frontal face images from the FERET database [11], excluding those images where the person wears glasses. The images used have been divided in two set: training and test with 60% and 40% of the images, respectively. It is worth noting that there are several images of the same person, but all of them are assigned to the same set of images.

The methodology design is divided in the following steps:

1. The face is detected using the Viola and Jones algorithm [6] implemented in the OpenCV [12] library. This algorithm is completely automatic since only takes the image as input. The system does not correct the inclination that the face might have.
2. The top half of the resulting image from step 1 (the area of the image where the face was detected) is extracted and then equalized and resized to a pre-established smaller size. The interpolation process required for the resizing step uses a three-lobed Lanczos windowed sinc function [13] which keeps the original image aspect ratio.
3. A set of windows of $N \times N$ pixels are defined to obtain a collection of vectors that characterize the top half of the face.
4. Given a test image, the classification process consists of assigning to each vector the class label (female or male) of its nearest neighbor in the training set. The gender of a test face is obtained by one of these procedures: 1) by majority voting of all the labels of the face's vectors or 2) by concatenating the vectors of all windows to build a longer vector to characterize the face, so the faces's class label will be the same as its vector's.

   The distance metrics used are the Euclidean metric and the Chi square metric and all the experiments have been done using both of them in order to compare which one performs better our task.

3.2 Description of the Classification Experiments

Four different experiments have been design to find out: 1) which is the face description that provides more information to discriminate between genders? and 2) which is the face description more suitable for situations where the face is not accurately detected?

The details about the experiments are presented next:

Experiment 1. In this case the top half face is split into a set of windows with no overlapping between them. This means that the pixels that belong to a window are never considered in another one. From each of the non-overlapping windows a feature vector is extracted. Then these vectors are concatenated to make a longer one. Hence, the vector of a certain window will be always compared with the vectors obtained from the windows that have the same position.

Experiment 2. In order to extract more detailed features to describe the top half face, overlapping windows are used in this case. Therefore, one pixel will belong to several windows and its value will be used to obtain the descriptions of all of them.

Although the size of the top half face images and the regions will be the same as in the previous experiment, the quantity of vectors will be higher because of the overlapping. Finally, all the vectors are also concatenated to make only one longer vector and, therefore, the features of a certain window will be always compared with the features obtained from the windows that have the same position in the training images.

Experiment 3. In this experiment the face is characterized by the same set of vectors obtained in experiment 2 but the classification process is different. Given a window of a test image, a set of neighboring windows will be considered in the training images. The size of that neighborhood depends on the error tolerance you may consider in the face detection process. The feature vector of the test window will be compared with the vectors obtained for all windows considered in the training images. The class label of the nearest neighbor is assigned to each window of the test face. Then, the test face obtains the class label resulting from the voting of all its windows. In our experiments the neighborhood considered is the whole face, so no precision is prefixed in the detection process. However, this approach leads to a high computational cost.

Due to the fact that each vector is individually classified and its nearest neighbor is not restricted to those vectors obtained from the regions in the same position, faces will not need to be accurately detected as in the previous experiments.

Experiment 4. This experiment presents a more realistic approach of the previous experiment. Now, the detection of the faces is artificially modified to simulate an inaccurate detection. The only difference with experiment 3 is that, after the automatic detection of the faces, a random displacement is applied to the area containing the face. The displacement could be at most 10% of the width for the horizontal movement and 10% of the height for the vertical one.

This experiment allows us to test the face descriptions and the classification methods in a more unrestricted scenario. Consequently, it could provide more reliable results about whether our system would be suitable for situations where the face detection could not be accurate.

3.3 Development

A complete set of experiments (see Table 1) have been carried out to test the face descriptions described in Sect. 2 and several combinations of them. Specifically, the face descriptions implemented are the following:

- Uniform LBP with neighborhoods of 8 sample points and radii 1 ($\mathrm{LBP}^{u}_{8,1}$) and 2 ($\mathrm{LBP}^{u}_{8,2}$).

- The combination of the $\text{LBP}^u_{8,1}$ + $\text{LBP}^u_{8,2}$ which consists in concatenating the vectors obtained with both descriptions.
- Local contrast histograms with neighborhoods of 8 sample points and radii 1 ($\text{LCH}_{8,1}$) or 2 ($\text{LCH}_{8,2}$).
- The combination of $\text{LCH}_{8,1}$ + $\text{LCH}_{8,2}$.
- The combination of LBP and LCH with the same number of sample points and radius. The resulting face descriptions are: $\text{LBP}^u_{8,1}$ + $\text{LCH}_{8,1}$ and $\text{LBP}^u_{8,2}$ + $\text{LCH}_{8,2}$.
- The combination of the two previous which results in $\text{LBP}^u_{8,1}$ + $\text{LCH}_{8,1}$ + $\text{LBP}^u_{8,2}$ + $\text{LCH}_{8,2}$.
- Ranking labels description.

All the face descriptions based on LBPs, produced two experiments: one with the sensitive to rotation version and the other one with the rotationally invariant version. In case of sensitive to rotation descriptions the vectors produced are composed of 10 features, while on the other case the vectors have 59 elements. Ranking labels description produces 49 features vectors.

In all the experiments, the amount of images used was 2147. The top half face images were reduced to a 45 × 18 pixels image. The size of the window that scans the images is 7 × 7 in all cases.

4 Results and Discussion

The correct classification rates obtained for each experiment carried out are shown in Table 1.

With regard to the distance metrics used, the Chi square succeeded in recognizing the genders with better rates than the Euclidean metric in 73% of the cases.

Concerning the radius of the neighborhood used for the histogram based features, radius 2 performs the recognition task better than radius 1 in 81% of the cases. Nevertheless, the combination of the same face description using both radii achieves higher rates, but using twice as many features.

As can be easily seen, the sensitive to rotation descriptions achieved better results than the rotationally invariant ones when only the LBPs are used. However, the use of 59-bin histograms to describe the LCH provided worse results in experiments 1 and 2. This could be explained by the higher dispersion of the data which leads to a poorer characterization which also causes lower recognition rates in most of the cases that combined LBP and LCH.

The results of experiments 1 and 2 show that the LCH is really useful to discriminate between genders since recognition rates reached by the LCH are very similar to those achieve using the LBP. LCH performs better when using rotationally invariant descriptions, whereas the rates obtained using LBP are slightly higher when the rotation dependent features were considered. As expected, when the LBP and the LCH with the radii 1 and 2 are used together to describe the faces, the recognition rate goes up until 82.69% (experiment 1)

Table 1. Recognition rates

		Experiment 1		**Experiment 2**		**Experiment 3**		**Experiment 4**	
		RI	no RI	RI	no RI	RI	no RI	RI	no RI
$\mathbf{LBP}^u_{8,1}$									
	χ_2	70.88	76.61	74.27	78.48	61.66	71.75	61.08	61.08
	Euclidean	68.30	76.02	73.33	76.37	61.08	70.57	61.08	61.08
$\mathbf{LBP}^u_{8,2}$									
	χ_2	68.42	79.06	81.17	78.95	61.43	75.26	61.08	61.08
	Euclidean	68.42	76.73	77.89	75.56	62.02	72.92	62.14	62.14
$\mathbf{LBP}^u_{8,1} + \mathbf{LBP}^u_{8,2}$									
	χ_2	73.92	80.47	78.13	80.23	62.84	78.55	62.49	62.49
	Euclidean	72.51	78.25	77.43	77.31	62.49	76.32	62.14	62.14
$\mathbf{LCH}_{8,1}$									
	χ_2	75.44	69.36	79.65	74.97	61.08	62.95	61.08	64.36
	Euclidean	73.57	70.64	78.95	72.87	61.08	64.36	61.08	65.06
$\mathbf{LCH}_{8,2}$									
	χ_2	77.89	71.81	79.77	75.79	61.08	63.42	61.08	63.19
	Euclidean	74.27	72.05	76.96	74.50	61.08	63.42	61.08	64.13
$\mathbf{LCH}_{8,1} + \mathbf{LCH}_{8,2}$									
	χ_2	77.89	72.98	79.30	76.26	65.06	64.48	64.83	65.30
	Euclidean	75.44	73.80	77.54	76.73	66.00	63.07	64.48	63.66
$\mathbf{LBP}^u_{8,1} + \mathbf{LCH}_{8,1}$									
	χ_2	75.79	79.53	80.23	81.17	66.47	79.95	64.83	79.01
	Euclidean	77.19	77.43	79.65	77.89	67.87	75.15	65.77	73.51
$\mathbf{LBP}^u_{8,2} + \mathbf{LCH}_{8,2}$									
	χ_2	80.47	79.88	**82.46**	81.40	69.05	82.65	69.17	81.71
	Euclidean	77.43	77.66	81.17	77.08	69.40	77.61	69.64	76.08
$\mathbf{LBP}^u_{8,1} + \mathbf{LCH}_{8,1} + \mathbf{LBP}^u_{8,2} + \mathbf{LCH}_{8,2}$									
	χ_2	**82.69**	81.64	82.81	80.82	74.44	85.11	71.16	83.59
	Euclidean	80.70	79.88	81.40	77.19	71.28	78.55	70.81	78.55
Ranking Labels									
	χ_2	78.95		80.12		**88.54**		89.12	
	Euclidean	78.60		79.30		**88.54**		**89.94**	

and 82.81% (experiment 2) which are the best rates of these experiment. However, the ranking label description achieved the best results when individual features were considered (not combinations of several features). To summarize, experiments 1 and 2 have proved that all the face descriptions are quite good to discriminate between genders. Not very important differences were obtained in the classification rates. In general, the more number of features used to describe the faces, the best classification rates obtained.

For experiments 3 and 4, the ranking labels description was the most suitable since it reached the best recognition rates which were close to 90%. That is, the correct classification rates were even better than for experiments 1 and 2. In our opinion, this is due to the fact that experiments 1 and 2 considered that the faces have always been perfectly located in the images. The error tolerance introduced in the classification experiments 3 and 4 helped to improve the rates obtained as they avoided the influence of the localization errors. However, this significant improvement only happens for the ranking labels features. Features based on individual histograms performed in these cases worse than for experiments 1 and 2. This is probably because the ranking label features keep the positional information of each pixel inside the corresponding window. Therefore, they keep their discriminative power even when the features of a certain window are compared against the features of another window which is located at a different spatial position. However, histogram based features required this correspondence between windows in the test and training images in order to keep their performance. Combining all histogram-based features, the classification rates also improved slightly, but using a very high number of features per window.

5 Conclusions

This paper has addressed the automatic gender classification problem in situations where the face was partially occluded and inaccurately detected.

The experiments have shown that LBPs and LCHs performed correctly when the positional information is kept by the classification method. However, these face descriptions are less reliable in situations with non-accurate face detections, since there is an important spatial information loss.

The best characterization method in an inaccurate environment was the ranking labels description which reached to almost a 90% of recognition rate due to the fact that these features were designed to keep the information about the position of the pixels in the different windows considered over the image.

Summing up, ranking labels are the most reliable characterization method as it performs in a similar way in all experiments carried out. Although, LBPs and LCHs performed correctly the gender recognition task, they were more dependent on the accuracy of the face localization process.

References

1. Rajagopalan, A.N., Rao, K.S., Kumar, Y.A.: Face recognition using multiple facial features. Pattern Recogn. Lett. 28(3), 335–341 (2007)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 2037–2041 (2006)
3. Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D., Rehg, J.M.: On the design of cascades of boosted ensembles for face detection. Int. J. Comput. Vision 77(1-3), 65–86 (2008)
4. Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(4), 671–686 (2007)
5. Garcia, C., Delakis, M.: Convolutional face finder: a neural architecture for fast and robust face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(11), 1408–1423 (2004)
6. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
7. Ahonen, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51–59 (1996)

8. Andreu, Y., Mollineda, R., García-Sevilla, P.: Gender recognition from a partial view of the face using local feature vectors. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.J. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 481–488. Springer, Heidelberg (2009)
9. Topi, M., Timo, O., Matti, P., Maricor, S.: Robust texture classification by subsets of local binary patterns, vol. 3, pp. 935–938 (2000)
10. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 971–987 (2002)
11. Phillips, H., Moon, P., Rizvi, S.: The FERET evaluation methodology for face recognition algorithms. IEEE Trans. on PAMI 22(10) (2000)
12. Bradski, G.R., Kaehler, A.: Learning OpenCV. O'Reilly, Sebastopol (2008)
13. Turkowski, K.: Filters for common resampling tasks, pp. 147–165 (1990)

Rigid Part Decomposition in a Graph Pyramid*

Nicole M. Artner[1], Adrian Ion[2], and Walter G. Kropatsch[2]

[1] AIT Austrian Institute of Technology, Vienna, Austria
nicole.artner@ait.ac.at
[2] PRIP, Vienna University of Technology, Austria
{ion,krw}@prip.tuwien.ac.at

Abstract. This paper presents an approach to extract the rigid parts of an observed articulated object. First, a spatio-temporal filtering in a video selects interest points that correspond to rigid parts. This selection is driven by the spatial relationships and the movement of the interest points. Then, a graph pyramid is built, guided by the orientation changes of the object parts in the scene. This leads to a decomposition of the scene into its rigid parts. Each vertex in the top level of the pyramid represents one rigid part in the scene.

1 Introduction

Tracking articulated objects and their rigid parts is an important and challenging task in Computer Vision. There is a vast amount of work in this field as can be seen in the surveys [1,2,3]. Possible applications are the analysis of human motion for action recognition, motion based diagnosis and identification, motion capture for 3D animation and human computer interfaces.

The first step in tracking articulated objects is the initialization. This step is important, because it can strongly influence the success of the tracking method. There are three possibilities for the initialization: (1) manually by the user, (2) solving the task as a recognition problem with the help of a training set [4], and (3) employing a segmentation method.

This paper presents an approach to segment the rigid parts of articulated objects from a video (third category). It is related to the concept of *video object segmentation* (VOS), where the task is to separate foreground from background in an image sequence. VOS methods can be divided into two categories [5]:

(1) Two-frame motion/object segmentation: Altunbasak et al. present in [6] a combination of pixel-based and region-based segmentation methods. Their goal is to obtain the best possible segmentation results on a variety of image sequences. Castagno et al. [7] describe a system for interactive video segmentation. An important key feature of the system is the distinction between two levels of segmentation: (1) regions and (2) object segmentation. Chen et al. [8] propose an

* Partially supported by the Austrian Science Fund under grants P18716-N13 and S9103-N13.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 758–765, 2009.

approach to segment highly articulated objects by employing weak-prior random forests. The random forests are used to derive the prior probabilities of the object configuration for an input frame.

(2) Multi-frame spatio-temporal segmentation/tracking: Celasun et al. [5] present VOS based on 2D meshes. Tekalp et al. [9] describe 2D mesh-based modeling of video objects as a compact representation of motion and shape for interactive video manipulation, compression, and indexing. Li et al. [10] propose to use affine motion models to estimate the motion of homogeneous regions.

There is related work explicitly dealing with the segmentation of articulated objects (e.g. Chen et al. [8]), but the result of these approaches is still only a separation of foreground and background. To initialize tracking methods for articulated object parts, it would be convenient to have a decomposition of the articulated foreground object into its rigid parts (e.g. for the method in [11]).

In this paper, we achieve the decomposition of the rigid parts of an articulated object. The scene is observed and analyzed over time. Depending on the spatial relationships and movements in the scene the input for the grouping process is selected. The grouping itself is done in a graph pyramid and is controlled by the orientation variation resulting out of the articulated movement of the object parts in the scene. This approach is a continuation of the work in [12].

The paper is organized as follows: Sec. 2 recalls graph pyramids. In Sec. 3 the spatio-temporal filtering is explained and Sec. 4 describes how the rigid parts are identify. Sec. 5 presents experiments and in Sec. 6 we give conclusions.

2 Irregular Graph Pyramids

A *region adjacency graph* (RAG), encodes the adjacency of regions in a partition. A vertex is associated to each region, vertices of neighboring regions are connected by an edge. Classical RAGs do not contain any self-loops or parallel edges. An *extended region adjacency graph* (eRAG) is a RAG that contains the so-called *pseudo edges*, which are self-loops and parallel edges used to encode neighborhood relations to a cell completely enclosed by one or more other cells [13]. The *dual* graph of an eRAG G is called *boundary graph* (BG) and is denoted by $\bar{G}$ (G is said to be the *primal* graph of $\bar{G}$). The edges of $\bar{G}$ represent the boundaries of the regions encoded by G, and the vertices of $\bar{G}$ represent points where boundary segments meet. G and $\bar{G}$ are planar graphs. There is a one-to-one correspondence between the edges of G and the edges of $\bar{G}$, which also induces a one-to-one correspondence between the vertices of G and the 2D cells (denoted by *faces*[1]) of $\bar{G}$. The dual of $\bar{G}$ is again G. The following operations are equivalent: edge contraction in G with edge removal in $\bar{G}$, and edge removal in G with edge contraction in $\bar{G}$.

A (dual) irregular graph pyramid [13,14] is a stack of successively reduced planar graphs $P = \{(G_0, \bar{G}_0), \ldots, (G_n, \bar{G}_n)\}$. Each level $(G_k, \bar{G}_k), 0 < k \leq n$

[1] Not to be confused with the vertices of the dual of a RAG (sometimes also denoted by the term *faces*).

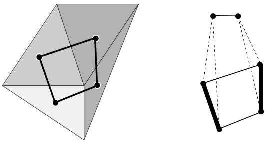


Fig. 1. Left: triangulation and associated adjacency graph. Right: contraction of two edges (thick) in a two level pyramid.

is obtained by first contracting edges in G_{k-1} (removal in $\bar{G}_{k-1}$), if their end vertices have the same label (regions should be merged), and then removing edges in G_{k-1} (contraction in $\bar{G}_{k-1}$) to simplify the structure. The contracted and removed edges are said to be *contracted* or *removed* in $(G_{k-1}, \bar{G}_{k-1})$. In each G_{k-1} and $\bar{G}_{k-1}$, contracted edges form trees called *contraction kernels*. One vertex of each contraction kernel is called a *surviving vertex* and is considered to have been 'survived' to $(G_k, \bar{G}_k)$. The vertices of a contraction kernel in level $k-1$ form the *reduction window* $W(v)$ of the respective surviving vertex v in level k. The *receptive field* $F(v)$ of v is the (connected) set of vertices from level 0 that have been 'merged' to v over levels $0 \dots k$.

For the sake of simplicity, the rest of the paper will only use the adjacency graph G, but for correctly encoding the topology, both G and $\bar{G}$ have to be maintained. Figure 1 shows an example triangulation and pyramid.

3 Spatio-temporal Filtering

The aim of the spatio-temporal filtering is to define the input for the grouping process. As mentioned in Sec. 1, the filtering is carried out by observing the scene in sequence of frames. This observation is realized by tracking interest points.

The filtering focuses on the spatial relationships of the interest points over time. A planar, triangulated graph G is used as representation for the filtering. The vertices V of the graph are the interest points and the edges E, which encode the spatial relationships, are inserted with a Delaunay triangulation [15].

As the aim is to find rigid parts of articulated objects, the task of the filtering is to select interest points corresponding to rigid parts. To identify these points, the changes of the edge lengths in the triangulated graph over time is considered. A triangle is *potentially rigid* for the decomposition process if its edges lengths do not vary remarkably in the observation period. In every frame of the video sequence the edge lengths $||e||$ can be calculated. To decide which triangles are *potentially rigid* the *edge length variation* is determined.

Definition 1. *The* ***edge length variation*** *of an edge is the difference between the minimum and the maximum length of the edge in the observation period.*

A triangle is labeled as *potentially rigid* if the edge length variations of all three edges are beneath a certain threshold ϵ. This threshold is necessary, because noise, discretization, and small imprecisions in the localization ability of the tracker[2] can affect the outputted positions of the interest points.

[2] e.g. the detection vs. localization problem in edge detection.

The result of the spatio-temporal filtering is a triangulation, where each triangle is labeled *potentially rigid* or *not rigid* (see Sec. 5 Fig. 2(b)). The *potentially rigid* triangles are then passed on as input to the grouping process (see Sec. 4).

4 Rigid Part Decomposition

The task of the rigid part decomposition is to split the *potentially rigid* triangles from the spatio-temporal filtering into groups of triangles, each describing one rigid part.

As this paper focuses on articulated objects, the triangles describe a locally deformable object, which follows a globally articulated motion. Due to the local deformation freedom the *edge length variation* of triangles belonging to a rigid part and those connecting such parts might not differ to much. E.g. even though the bones of a human perform an articulated motion, the flesh and skin are elastic and thus a smooth and continuous deformation can be observed in the triangles going from the lower arm to the upper arm and then to the torso. Another aspect is that if the region around an articulation point is densely sampled (tracked by many points), the *edge length variation* resulting out of a rotation of e.g. 90 degrees is going to be insignificant. For all these reasons, the *edge length variation* is not a sufficient property for the decomposition task.

The idea is to group triangles into rigid parts where all the triangles inside a group have a similar *orientation variation* over the whole video, and the average orientation variation of the triangles in two neighboring groups differs. This problem is similar to the single image segmentation problem, where the results should be regions with homogeneous color/texture (small internal contrast) neighbored to regions that look very different (high external contrast).

Definition 2. *The **orientation variation** over time is a 1D signal that encodes at each time step (frame of the input video) the accumulated orientation change relative to the orientation at the beginning of the video.*

E.g. turning around the axis once will give a value of 360° degrees, and turning twice will give 720°, not 0°. The direction of rotation is encoded by the sign: counter clockwise (CCW) is positive, and clockwise (CW) is negative, e.g if turning 30° CCW, then 15° CCW, and then 28° CW, the computed variations will be 30°, 45° = 30°+15°, 17° = 45°-28°.

Definition 3. *The **orientation variation of a triangle** is the 1D signal obtained by taking the average of the 1D signals of the three edges of the triangle.*

Using an irregular pyramid for the grouping task has the advantage that its structure adapts to the data. Also, using a hierarchy reduces the complexity of the grouping (global decisions become local ones), and the produced description contains information that can be used to cope with complexity in a coarse-to-fine tracking approach.

Alg. 1 creates a graph pyramid in which each top level vertex identifies a detected rigid part. The receptive field of these vertices identifies the triangles

Algorithm 1. *BuildPyr*(T): Group triangles into rigid parts.

```
Input: potentially rigid triangles $T$ (see Section 3)
1: $G_0 = (V_0, E_0)$
   /*$V_0 = T$, and $(v, w) \in E_0 \iff$ the corresponding triangles share an edge*/
2: $k = 0$
3: repeat
4:    /*select edges to contract*/
      $K = \emptyset$
      $\forall v \in G_k$ do $K \leftarrow K \cup \arg\min_{(v,w)\in G_k}\{X(v, w)\}$
5:    /*filter edges based on internal/external difference*/
      $\forall (v, w) \in K$, if $X(v, w) > I'(v, w)$ then $K \leftarrow K \setminus \{(v, w)\}$
6:    if $K \neq \emptyset$ then break $K$ into trees of radius 1
7:    if $K \neq \emptyset$ then $G_{k+1} \leftarrow$ contract edges $K$ in $G_k$ and simplify
8:    $k \leftarrow k + 1$
9: until $K = \emptyset$
Output: Graph pyramid $P = \{G_0, \ldots, G_{k-1}\}$.
```

that the respective part consists of. In the base level, one vertex is associated to each *potentially rigid* triangle. Two vertices are connected by an edge if the respective triangles share a common edge. Edges to be contracted are selected from the edges proposed by the Minimum Spanning Tree Alg. by Boruvka [16] (Line 4). The **external difference** $X(v, u)$ between two vertices is:

$$X(v, w) = \max(|V(v) - V(w)|) \tag{1}$$

where $V(v), V(w)$ are the 1D signals associated to v respectively w. They encode the average of the orientation variation of the triangles in the receptive fields. For the vertices in the base level G_0 they are the orientation variations of the corresponding triangles. For a vertex in a higher level they can be computed as:

$$V(v) = \frac{\sum_{u\in W(v)} |F(u)| \cdot V(u)}{\sum_{p\in W(v)} |F(p)|} \tag{2}$$

where $|F(v)|$ is the size of $F(v)$ and can be propagated up in the pyramid. The **internal difference** $I(v)$ of a vertex at level $k > 0$ is:

$$I(v) = \max(\max\{I(u)\}, \max\{X(p_i, p_j)\}) \tag{3}$$

where $u \in W(v)$ and $p_i, p_j \in W(v)$ s.t. p_i, p_j are connected by an edge. For the vertices in the base level $I(v) = 0$. The value $I'(v, w)$ is defined as:

$$I'(v, w) = \min(I(v) + \frac{\beta}{|F(v)|}, I(w) + \frac{\beta}{|F(w)|}) \tag{4}$$

where β is a parameter of the method that allows regions to start forming in the base level where $I(v) = 0$ for all vertices. For a discussion about β in the context of image segmentation see [17,18].

Line 6 of Alg. 1 keeps the contraction operations local (optimal for parallel processing) and avoids contracting the whole graph in a single level. It excludes edges from K to create trees of radius 1 for the current contraction. The excluded edges will be selected again in the next level. In [19] three such methods, MIES, MIS, and D3P (used in our experiments) are described. The described concept is related to the image segmentation method in [17], with the difference that:

1. we do not start form a pixel image, but from a triangulation;
2. our features are not color values but 1D signals of orientation variation;
3. and most important, we do not assign the edge weights based on the weights in the level below, but recompute them to reflect the difference between the average variation of the triangles in the two neighboring regions.

The difference at 3. has the effect that a long chain of regions that differ by a constant, small difference, will not be merged to create a single region.

5 Experiments

The moving foreground objects in the experiments are humans, but the presented approach is applicable to any arbitrary object. The Kanade-Lucas-Tomasi tracker [20] is used to track corner points to supply the necessary observations.

Both video sequences show a person undergoing articulated motion in the image plane. In Fig. 2 the result of the triangulation and the following spatio-temporal filtering are visualized. Fig. 3 presents the decomposition results. The result for sequence 1 is ideal, meaning that each rigid part and the background are one vertex in the top level of the graph pyramid. For sequence 2 the right lower arm is represented by two top vertices. Also, two additional regions, one corresponding to a part of the hair, and one connecting the left upper arm with the background have also been produced as rigid parts. The reason for this are

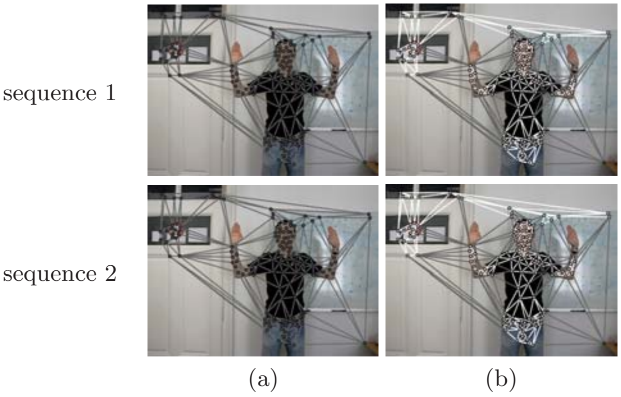


Fig. 2. Triangulation (a) without (b) with labeling. White: potentially rigid, grey: not rigid.

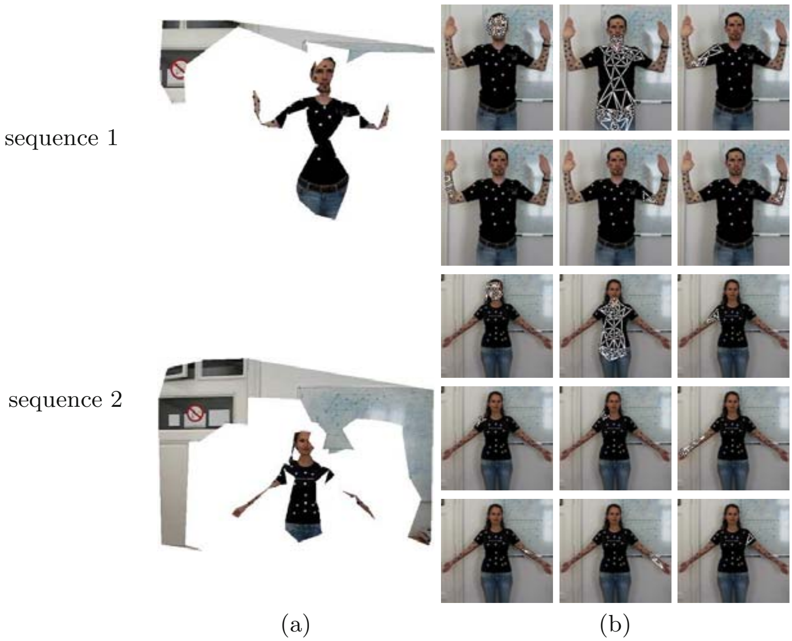


Fig. 3. (a) pixels belonging to a rigid part. (b) graphs for rigid parts of the foreground.

the difficulties mentioned at the beginning of Sec. 4 and the fact that the labeling into *potentially rigid* and *not rigid* has to allow certain variation (see Sec. 3). The torso is connected with the base of the chin in both sequences because during tracking the features at the base of the chin slide when the head is tilted and remain in the same position in the image, creating a *potentially rigid* triangle.

6 Conclusion

This paper presented a graph-based approach to decompose the rigid parts of articulated objects. First a spatio-temporal filtering is performed, where the spatial relationships of the interest points over time are analyzed and a triangulation is produced, with triangles labeled as *potentially rigid* and *not rigid*. The *potentially rigid* triangles are given as input to a grouping process that creates a graph pyramid s.t. in the top level each vertex represents a rigid part in the scene. The orientation variation of the input triangles controls the building process of the pyramid and is used to compute the similarity between two groups of triangles. The success of the presented approach depends on the quality and robustness of the tracking results. The presented approach fails if there are remarkable perspective changes or scaling. This is one of the open issues we are planing to deal with in the future. Additionally, we are going to find the articulation points connecting the rigid parts of the foreground objects.

References

1. Gavrila, D.M.: The visual analysis of human movement: A survey. CVIU 73(1), 82–980 (1999)
2. Moeslund, T.B., Hilton, A., Krger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104(2–3), 90–126 (2006)
3. Aggarwal, J.K., Cai, Q.: Human motion analysis: A review. CVIU 73(3), 428–440 (1999)
4. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61(1), 55–79 (2005)
5. Celasun, I., Tekalp, A.M., Gokcetekin, M.H., Harmanci, D.M.: 2-d mesh-based video object segmentation and tracking with occlusion resolution. Signal Processing: Image Communication 16(10), 949–962 (2001)
6. Altunbasak, Y., Eren, P.E., Tekalp, A.M.: Region-based parametric motion segmentation using color information. Graphical Models and Image Processing 60(1), 13–23 (1998)
7. Castagno, R., Ebrahimi, T., Kunt, M.: Video segmentation based on multiple features for interactive multimedia applications. Circuits and Systems for Video Technology 8(5), 562–571 (1998)
8. Chen, H.T., Liu, T.L., Fuh, C.S.: Segmenting highly articulated video objects with weak-prior random forests. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 373–385. Springer, Heidelberg (2006)
9. Tekalp, A., Van Beek, P., Toklu, C., Gunsel, B.: Two-dimensional mesh-based visual-object representation for interactive synthetic/natural digital video. Proceedings of the IEEE 86(6), 1029–1051 (1998)
10. Li, H., Lin, W., Tye, B., Ong, E., Ko, C.: Object segmentation with affine motion similarity measure. Multimedia and Expo, 841–844 (2001)
11. Artner, N., Ion, A., Kropatsch, W.G.: Tracking objects beyond rigid motion. In: Workshop on Graph-based Representations in PR, May 2009. Springer, Heidelberg (2009)
12. Mármol, S.B.L., Artner, N.M., Ion, A., Kropatsch, W.G., Beleznai, C.: Video object segmentation using graphs. In: 13th Iberoamerican Congress on Pattern Recognition, September 2008, pp. 733–740. Springer, Heidelberg (2008)
13. Kropatsch, W.G.: Building irregular pyramids by dual graph contraction. Vision, Image and Signal Processing 142(6), 366–374 (1995)
14. Kropatsch, W.G., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. PRL 26(3), 319–337 (2005)
15. Tuceryan, M., Chorzempa, T.: Relative sensitivity of a family of closest-point graphs in computer vision applications. Pattern Recognition 24(5), 361–373 (1991)
16. Nesetril, J., Milková, E., Nesetrilová, H.: Otakar boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. Discrete Mathematics 233(1-3), 3–36 (2001)
17. Haxhimusa, Y., Kropatsch, W.G.: Segmentation graph hierarchies. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 343–351. Springer, Heidelberg (2004)
18. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
19. Kropatsch, W.G., Haxhimusa, Y., Ion, A.: Multiresolution Image Segmentations in Graph Pyramids. In: Applied Graph Theory in Computer Vision and Pattern Recognition. SCI, vol. 52, pp. 3–42. Springer, Heidelberg (2007)
20. Birchfeld, S.: Klt: An implementation of the kanade-lucas-tomasi feature tracker (March 2008), http://www.ces.clemson.edu/~stb/klt/

Multimodal Algorithm for Iris Recognition with Local Topological Descriptors*

Sergio Campos[1], Rodrigo Salas[2], Hector Allende[1], and Carlos Castro[1]

[1] Universidad Técnica Federico Santa María, Dept. de Informática, Valparaíso-Chile
scampos@inf.utfsm.cl, hallende@inf.utfsm.cl, ccastro@inf.utfsm.cl
[2] Universidad de Valparaíso, Departamento de Ingeniería Biomédica,
Valparaíso-Chile
rodrigo.salas@uv.cl

Abstract. This work presents a new method for feature extraction of iris images to improve the identification process. The valuable information of the iris is intrinsically located in its natural texture, and preserving and extracting the most relevant features is of paramount importance. The technique consists in several steps from adquisition up to the person identification. Our contribution consists in a multimodal algorithm where a fragmentation of the normalized iris image is performed and, afterwards, regional statistical descriptors with Self-Organizing-Maps are extracted. By means of a biometric fusion of the resulting descriptors, the features of the iris are compared and classified. The results with the iris data set obtained from the Bath University repository show an excellent accuracy reaching up to 99.867%.

Keywords: Iris recognition, SOM, Voronoi polygons, regions descriptors.

1 Introduction

In recent years, the use of biometric systems, mainly for reasons of security, has grown almost exponentially. The authentication of the person's identity in an univocal and automatic way is a requirement by nowadays standards, even more due to large-scale applications that work with hundreds or thousands of users. For these reasons, it is necessary to improve the algorithms to expand the possible scenarios where these systems can be applied.

Biometric systems are commonly employed in verification mode, i.e., they work verifying the identity of the user by fitting it with the identity previously stored by any device used for accurate identification (card ID, password, etc.). This functionality is indispensable by banks, hospitals, government institutions, airports, to name a few possible applications. Moreover, it is also necessary to consider scenarios where we have no a priori information about the subject. Under this scene, the biometric system can work in identification mode, which without any information, and only with an input, searches within the database

* This work was supported by the Fondecyt 1070220 and 1070268 research grants and DGIP-UTFSM grant.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 766–773, 2009.

and tries to fit that input with the most similar user. This way of working is complicated when the system has more users, since it has to deal with the problem of reducing intra-class variance and increase the inter-class variance.

In this paper we propose a new method for iris recognition, which works the iris image as a multimodal system by fragmenting the image in two sectors. With an unsupervised method we obtain regions descriptors for each fragment, where this features preserves and compresses the iris texture information. Theses features are used to classify the images as identification mode.

The structure of our work is the following: section 2 shows a state of art of iris recognition; in section 3 we state our proposal of a new method, starting from the pre-processing up to the classification process; section 4 shows the results of the experiments, and section 5 gives some concluding remakrs about this work.

2 Related Works on Iris Recognition

John Daugman[1] proposed the first known algorithm for iris recognition. The process basically consists in the following steps. First, the pre-processing stage detects the edges of the pupil and iris to locate the position of the iris within the image. This stage is known as pre-processing. Then the process continues with the extraction of a pattern of the iris image by methods of feature extraction and finally performs the classification process.

The literature focuses mainly on creating new methods for these 3 major processes (pre-processing, feature extraction and classification). Daugman uses an integrated operator differential, two dimensions Wavelets and matching using XOR function for this processes respectively.

There are many other jobs that are based on Daugman's job, but they used different techniques, for example, in [8] they used independent component analysis (ICA) for feature extraction. In this paper the authors state that their method creates a more compact *iriscode*[1] than the more classical approaches and therefore the matching is faster. In addition, this process use only a portion of the iris, as in [5], but these use Gabor Filters.

Since 2005 approximately, various learning algorithms have been employed for this purpose, in [10] a proposal based on HSOM (a variant of SOM (*Self-Organizing Maps*) to hold the Hamming distance) is presented to calculate the adjustment of the patterns obtained in the process of feature extraction, in [9] an algorithm based in LVQ (*Learning Vector Quantization*) is presented to classify the patterns extracted previously through two dimensions Wavelets transform.

In recent years researchs oriented to biometric recognition have been focused on constructing algorithms for multimodal systems. Multimodal systems are those who work with multiple sources of information to make a decision, for example, several samples of the fingerprint, both iris of a person, different algorithms that work independently and deliver different outcomes, etc[6]. The fusion process is highly considered because the how and where the fusion takes place has a direct influence on the systems performance. Daugman notes the importance of standardizing the process of fusion and some methods to make the fusion at decision level, as AND and OR rule[4].

Generally to work with more sources of information should improve the classification rates, which is shown in some studies as [11] y [12], where the first involves iris and face and attacks the problem of score fusion as an optimization problem, where the task is to find the threshold which minimizes the total error, and the second which work with 3 traits: iris, face and palms, which performs fusion at score level, through multiple SVMs in parallel, but this only works if good classifiers are used and/or the information source is not noisy.

3 The Proposed Method

3.1 Localization

The location is primarily to recover the portion of the iris in the image. For this purpose, we proceeded to find a preliminary center of the pupil (x_p, y_p) by checking the vicinity of each pixel, which should have a low intensity and be central to the image. After establishing the point (x_p, y_p), we proceed to find the real center (x_c, y_c) through a square that circunscribe the pupil, detected by the difference of intensity between the pupil and the iris.

After obtaining the center of the pupil, we proceed to detect the inner boundary (border) of the iris and with it, the radius of the pupil r_p.

Finally, we use the procedure described in [3], which creates a circle of radius R $(r_p < R)$ concentric to the pupil that is used to clean the image. Thus, the portion of the iris considered for the recognition is the area located in between the pupil and the circle of radius R. Everything outside this circle is eliminated by reseting the pixeles to 0.

3.2 Normalization

It is necessary to extract the texture information of the iris of the original image once it is fully identified, so we can work with it more easily. We use the Polar transform [5], which is based on Daugman's work [1]. The difference lies mainly in the sweep angle. In [5], the authors took a sweep angle of 90 for the left and right section of the iris, not considering the upper and lower sections of the iris, because of the possible occlusions produced by eyelashes.

Due to the characteristics of the images in the database[13], we modify the polar transform proposed in [5], written in the following way:

$$J(x, y) = IE(x_0 + r\cos(\theta), y_0 + r\sin(\theta)) \tag{1}$$

where:

$$r = r_p + (x-1)\Delta_r \qquad \forall x \in N : x \leq \frac{r_i - r_p}{\Delta_r}$$

$$\theta = \begin{cases} ang\_1 + (y-1)\Delta_\theta & \text{, if } y \leq \frac{ang\_bar}{2\Delta_\theta} \\ \\ ang\_2 + (y-1)\Delta_\theta & \text{, if } y > \frac{ang\_bar}{2\Delta_\theta} \end{cases} \quad \forall y \in N : y \leq \frac{ang\_bar}{\Delta_\theta}$$

where (x_0, y_0) are the coordinates of the center of the pupil, r_p and r_i are the values of the radius of the pupil and the iris, respectively, Δ_r is the interval of separation between pixels of the same radio (if $\Delta_r = 1$ means that there is no separation), Δ_θ is the interval of separation angle between radio and radio, *ang_1*, *ang_2* are starting angles to make sweeping and *ang_bar* is the total angle covered by the full sweep of the process. The result of this process can be seen in Figure 1.

3.3 Enhancement of the Image

The normalized iris image has low contrast and could have a non-uniform brightness due to the light of the moment when the image was obtained. This makes the iris texture to be more uniform than it really is. This is why we performed an improvement to the image by means of a histogram equalization, which makes an expansion of the histogram of the image, ie, it makes it fill the full spectrum of shades of gray, hence increasing the contrast in such a way that the texture patterns are easily noted. Figure 1 shows the process.

3.4 Fragmentation of the Image

The process of fragmentation is applied to the normalized and enhanced image. The idea is to divide this image into N smaller fragments of the same size, to reduce the computational complexity of the process of feature extraction. The figure 1 shows an example of fragmentation with $N = 2$. Note that each fragment is considered as if they were different iris but of the same subject, which allows us to perform fusion.

3.5 Construction of the Topological Graph

The iris texture is extremely rich in detail, and as mentioned above, the shape and distribution of these items is unique in each iris. To rescue the pattern obtained from the texture of the iris, we proceeded to construct a graph which preserve the topology of the iris in the graph. This graph will be the template for each iris image, which has the sufficient information to make a direct fit with the images considered in-put, ie, carry out a direct fit between the topological graph and image.

For the construction of the topological graph a neural network SOM was used. Before handing the input to the network is performed a binarization of the image using the Otsu method[7], which chooses a threshold that minimizes the variation of the intra-class white and black pixels in the image.

With the application of the SOM to the binarized image, the most relevants patterns are identified by the resulting graph. The training vectors of the network SOM are the coordinates (x_i, y_i) of the black pixels of the image. The neurons of the grid are positioned such that they describe the distribution of black pixels. The figure 2 shows the process of feature extraction.

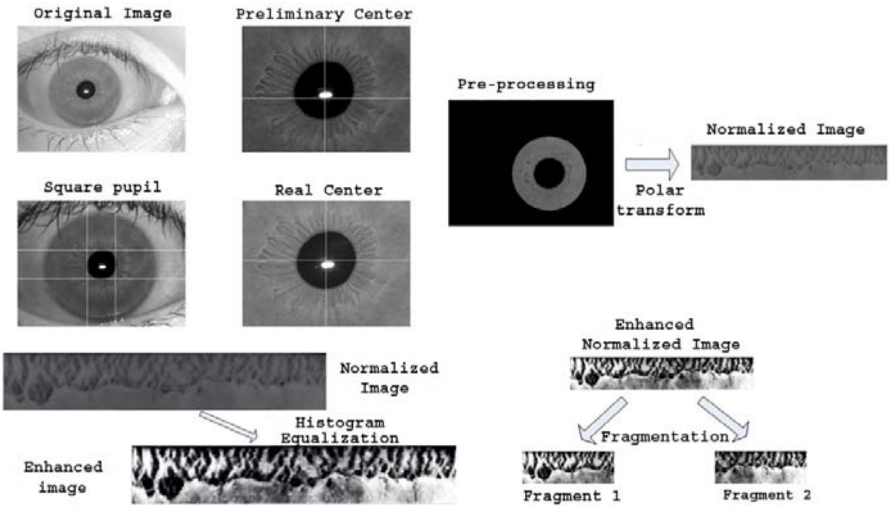


Fig. 1. Pre-processing process

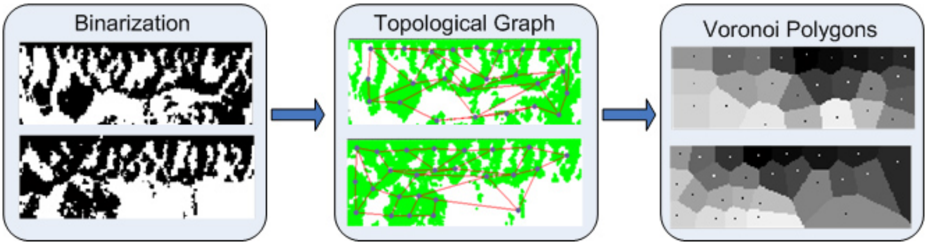


Fig. 2. Feature extraction process

3.6 Region Statistical Descriptors

Once the graph is constructed, the next step consists in obtaining the set of regional statistical descriptors. For this purpose, each node in the topological graph identify their areas of influence using Voronoi polygons.

Consider a finite number, n, of points in the Euclidean plane and assume that $2 \leq n < \infty$. The n points labeled by $p_1, p_2, ..., p_n$ with Cartesian coordinates $(x_{11}, x_{12}), (x_{21}, x_{22}), ..., (x_{n1}, x_{n2})$. The n points are different in the sense that $p_i \neq p_j$ to $i \neq j$, $i, j \in I_n = [1, 2, ..., n]$.

Let p be an arbitrary point in the Euclidean plane with coordinates (x_{p1}, x_{p2}). Then the Euclidean distance between p and p_i is given by $d(p, p_i) = \|x_p - x_i\| = \sqrt{(x_{p1} - x_{i1})^2 + (x_{p2} - x_{i2})^2}$. If p_i is the point closest to p, then the relationship $\|x_p - x_i\| \leq \|x_p - x_j\|$ to $i \neq j$, $i, j \in I_n$ holds. In this case, p is assigned to p_i. So, mathematically we can define a Voronoi diagram as follows [2]:

Let $P = [p_1, p_2, ..., p_n] \subset \mathbb{R}^2$, where $2 \leq n < \infty$ and $x_i \neq x_j$ to $i \neq j$, $i, j \in I_n$. Let the region given by: $V(p_i) = \{x| \ \|x - x_i\| \leq \|x - x_j\| \ for \ j \neq$

$i, j \in I_n\}$ *planar Voronoi polygon* associated with p_i, and the set given by: $\boldsymbol{V} = \{V(p_1, V(p_2), ..., V(p_n))\}$ *planar Voronoi diagram* generated by P. After identifying the areas of influence of each node, we proceed to calculate statistical descriptors of the regions.

The statistical descriptors calculated were the mean, variance and skewness. The first 3 moments have a great descriptive power in terms of intensity distribution of pixels covered. In this way, for every region of influence the first 3 moments were calculated, normalized between 0 and 1 (only μ and σ) as follows: $\mu = \frac{1}{255}\frac{1}{n}\sum_{i=1}^{n} x_i$, $\sigma = \frac{1}{\mu \cdot 255^2}\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$, and $\gamma = \frac{E(x-\mu)^3}{\sigma^3}$ obtaining a vector of features as follows: $\boldsymbol{V} = (\mu_1, \sigma_1, \gamma_1, ..., \mu_{25}, \sigma_{25}, \gamma_{25})$.

3.7 Classification

For the classification process we use the vectorial similarity function that measures how similar are 2 images:

$$F_{sim} = |\,\boldsymbol{V}_{in} - \boldsymbol{V}_t| \tag{2}$$

where $\boldsymbol{V}_{in}$ is the vector of characteristics of the input image and $\boldsymbol{V}_t$ is the template vector previously stored in the database.

4 Experimental Results

For our experiments, the database employed was obtained from the University of Bath repository[13].The data set consists of 1000 images of 25 different subjects with 20 images for each eye. The images are in grayscale and an original size of 1280 by 960 pixels. The computer where the experiments were conducted was a Pentium 4 1.5 Ghz with 1 GB of RAM. The experiments were done in Identifcation mode, that is, given an input image, it was classified according to the closest class of the total database. This mode only generates one error rate, unlike the Verification mode that generates 2 types of error: False Accept rate (FAR) y False Reject Rate (FRR)[6].

Due to the difference in the patterns of the iris of the left and right eye of each person, we considered each eye as a different class, so the total number of classes that worked were 50. For each iris we got 2 fragments which we used to get the score level fusion.

The experiments were performed with 5-fold cross validation. The fusion was performed using a scoring rule that considers the following weights for each fragment:

$$Score_t = 0.5 \cdot score_{frag1} + 0.5 \cdot score_{frag2} \tag{3}$$

where $Score_t$ is the total score for the classification.

Table 1 shows the accuracy rate , variance and time obtained in the identification proceses for with fragmentation and without fragmentation. Three different configuration of statistical descriptors were tested: the mean, the mean and the

Table 1. Summary of experimental results

	Without Fragmentation			**With Fragmentation**		
Descriptors	**Acc.** [%]	**Var.** [%]	**Time** $[\frac{sec}{match}]$	**Acc.**[%]	**Var.**[%]	**Time**$[\frac{sec}{match}]$
μ	99,655	0,066	0,095	**99,867**	0,026	**0,077**
μ and σ	99,517	0,104	0,113	99,203	0,102	0,083
μ, σ and γ	99,787	0,056	0,122	99,787	0,005	0,109

variance, and the mean, variance and skewness. The best outcome was obtained for the mean reaching up to 99.867%.

While it is logical to expect that including more features will describe in a better way the processed image, the addition can deteriorate the performance of the classification because, probably they do not supply any distinctive information for each class or combined with another occurrence of the classes overlap, that is, does not increase the inter-class variance.

With respect to time, it clearly wins by getting 0.077 seconds by each comparison, which apparently is not significant when compared with the third configuration's 0.109 seconds, but if we think in large scale, i.e. thousands of users, the improvement would be in the order of ten minutes.

The main goals of our proposed algorithm is twofold: first, it was to reduce the dimension of the images of the iris and in this way to reduce the complexity for the feature extraction process. The second was to have the opportunity to make a single image fusion without having the original pattern splitted, and therefore, work with only one part of the iris image.

5 Conclusions and Further Works

In this work, a new method for iris recognition were presented. This method allows us to recover the topology of the iris through the neural network SOM, technique which gives us important points within the iris image represented by the neurons. Thanks to these points, we can identify areas of infuence of each neuron using Voronoi polygons, and also characterize these polygons using regions descriptors.

The techniques employed in our method describe very well the texture of the iris, which is defined as a sub-structure and each of these are quantified according to the distribution of intensities of the pixels.

The results of the experiments show that the fractionation of the image enhances the rate of accuracy, to almost 100 %. This tells us that it is not necessary to have more sources of information, in order to improve the rate of accuracy which involves more complex computational, but if each fragment separately allows a good classification, together will surely improve overall performance[4].

Our future work will focus primarily on 2 tasks to develop: be calculated more regions descriptors, such as a translations and rotations invariant moments, entropy, etc., and all these will develop a feature selection process to know which are the best descriptors in terms of it's descriptive quality and computational complexity. As a second task, since we are interested in working in Identification mode, perform clustering of users by the topological graph, well to address environments to deal with so many users. For this work with more than one database.

References

1. Daugman, J.: How Iris Recognition Works. IEEE Trans. on Circuits and Systems for Video Technology 14, 21–30 (2004)
2. Okabe, A., Boots, B., Sugihara, K., Chiu, S.: Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, segunda edición. John Wiley & Sons Ltd, Chichester (2000)
3. Ganeshan, B., Theckedath, D., Young, R., Chatwin, C.: Biometric iris recognition system using a fast and robust iris localization and alignment procedure. Optics and Lasers in Engineering 44, 1–24 (2006)
4. Daugman, J.: Biometric Decision landscape. Technical report 482, University of Cambridge (2000)
5. Sanchez-Reillo, R., Sanchez-Avila, C.: Iris recognition with Low Template Size. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 324–329. Springer, Heidelberg (2001)
6. Jain, A., Ross, A., Nandakumar, K.: Handbook of Multibiometrics. Springer, Heidelberg (2006)
7. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics 9(1), 62–66 (1979)
8. Noh, S., Bae, K., Park, K., Kim, J.: A new Iris recognition method using independent component analysis. IEICE Trans. Inf. And Syst E88-D (2005)
9. Cho, S., Kim, J.: Iris Recognition using LVQ Neural Network. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 26–33. Springer, Heidelberg (2006)
10. Neagoe, V.: New Self-Organizing Maps with Non-conventional metrics and their applications for Iris recognition and automatic translation. In: Proceedings of the 11th WSEAS International Conference on Computers (2007)
11. Toh, K., Kim, J., Lee, S.: Biometric scores fusion based on total error rate minimization. Patter Recognition 41, 1066–1082 (2008)
12. Wang, F., Han, J.: Robust Multimodal Biometric Authentication Integrating Iris, Face and Palmprint. 124X Information Technology and Control 37(4) (2008)
13. Monro, D.: Bath University iris database (2008), http://www.bath.ac.uk/elec-eng/research/sipg

Scene Retrieval of Natural Images

J.F. Serrano[1], J.H. Sossa[1], C. Avilés[2], R. Barrón[1], G. Olague[3], and J. Villegas[1]

[1] Centro de Investigación en Computación-Instituto Politécnico Nacional (CIC- IPN)
UPLM – Zacatenco, Av. Juan de Dios Bátiz y Othón de Mendizábal s/n Col. Lindavista
C.P. 07738 México, D. F. Tel.: 57296000 ext. 56588
[2] Universidad Autónoma Metropolitana-Azcapotzalco. Departamento de Electrónica,
Av. San Pablo 180, Col Reynosa, México D.F. C.P. 02200, Tel.: 53189550 ext 1026
[3] Centro de Investigación Científica y de Educación Superior de Ensenada, BC.
Km 107 Carretera Ensenada-Tijuana, No. 3918, Zona Playitas, C.P. 22860, México
jfserranotal@gmail.com, hsossa@cic.ipn.mx,
caviles@correo.azc.uam.mx, rbarron@cic.ipn.mx,
gustavo.olague@gmail.com, jvillegas@gmail.com

Abstract. Feature extraction is a key issue in Content Based Image Retrieval (CBIR). In the past, a number of describing features have been proposed in literature for this goal. In this work a feature extraction and classification methodology for the retrieval of natural images is described. The proposal combines fixed and random extracted points for feature extraction. The describing features are the mean, the standard deviation and the homogeneity (form the co-occurrence) of a sub-image extracted from the three channels: H, S and I. A *K*-MEANS algorithm and a 1-NN classifier are used to build an indexed database of 300 images. One of the advantages of the proposal is that we do not need to manually label the images for their retrieval. After performing our experimental results, we have observed that in average image retrieval using images not belonging to the training set is of 80.71% of accuracy. A comparison with two similar works is also presented. We show that our proposal performs better in both cases.

1 Introduction

Nowadays, due the availability of large storage spaces a huge number of images can be found in the Internet. With this huge distributed and heterogeneous image database, people want to search and make use of the images there contained. A great challenge emerges: finding out accurate ways of searching images. Basically, images can be retrieved in two ways, firstly, text based and secondly, content-based or query by example based. Text-based retrieval approaches are very well-known and widely used. In this case users are provided with a text area to enter the key words (usually the image file name) on the basis of which image searching is done. It is widely used in Google web based image searching technique.

The concept CBIR has a main drawback: The images in the database are manually annotated using key words. This is known to be a very time consuming process for any large database [1], [2]. Also retrieval depends on the human perception based text annotation.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 774–781, 2009.

To avoid the above mentioned problems, a second approach, Content-Based Image Retrieval (CBIR) has been proposed by researchers. The term CBIR seems to have originated in the earlier 90´s [1], [4], [5], [6], [10], [12], [14] and [15]).

CBIR includes research on: Automatic Feature Extraction ([2], [3]), Automatic Feature Extraction with a Semantic Content ([4], [5], [6], [9], [10] and [11]) and data representation ([7]). CBIR techniques use low-level features such as texture, color and shape to represent images and retrieves images relevant to the query image from the image database. Among those low level image features, texture features has been shown very effective and subjective [15].

In this paper we describe a CBIR based methodology. In the next section we describe each of the steps composing the proposed approach.

2 Methodology

In this section we describe each of the stages of the proposed methodology for the retrieval of natural images into a database. It involves two basic stages as follows:

Training stage. This stage is divided into two main phases as shown in Fig. 1(a). During the first phase a set of 300 images in RGB format is first read. Each image is converted to HSI format. To each image, 300 pixels are uniformly selected at random (see Fig. 2(a)). Taking each of the 300 points as the center we open a squared window of size of 10×10 pixels around it.

Figure 2(b) shows several examples. To each of the 300 windows the following features are extracted: the mean, the standard deviation [13] and the homogeneity obtained from the co-occurrence matrix [8]. This is done for the corresponding

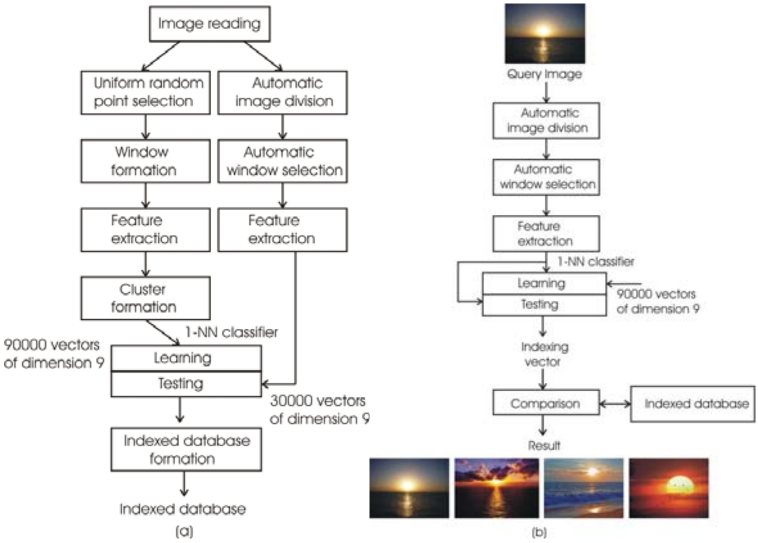


Fig. 1. (a) Flow Diagram for the training stage. (b) Flow diagram of the testing stage.

sub-image channel: hue (H), saturation (S) and brightness (I) of an image. The corresponding describing vector for each window of the image has thus nine components, three for H channel, three for S channel and three for I channel.

We take the resulting 90,000 describing vectors (300 for each of the 300 images) and a *K*-MEANS algorithm is applied to obtain how many of these 90,000 features are divided into six classes. For the 300 images chosen in this paper for training, Table 1 shows how many vectors fall into class one, how many vectors fall into class two, and so on until class six. This gives somehow the probability that a given class belongs to the 300 images.

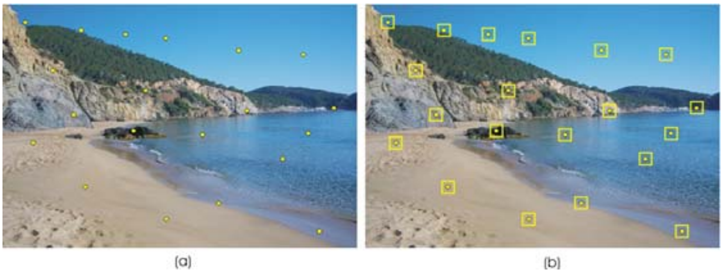


Fig. 2. (a) For sub-image description 300 image pixels are automatically and uniformly selected at random. (b) For automatically image segmentation around each of the 300 pixels a square window of *M*×*N* is opened. In this figure only 20 points are shown as an example.

Table 1. Distribution of the 90,000 features into the 6 chosen classes

Class number	Number of features per class
1	14,647
2	16,106
3	7,104
4	19,155
5	11,848
6	21,140
	Total 90,000

During the second phase, to the same set of 300 images an automatic partition is performed as shown in Fig. 3(a). As shown in this figure each image is divided into 10×10 regions of 72×48 pixels per region. For each of these 100 sub-images we take a window of 10×10 pixels as shown in Fig. 3(b). To each of the resulting 100 windows, again the same: mean, standard deviation and the homogeneity are computed in the three same channels. Each window is described again in the form of vector of nine components. As a result we have 30,000 vectors (100 for each of the 300 images).

To create the indexed database of the 300 images used for training we proceed as follows. We take the 90,000 describing vectors obtained in the first phase of training and the 30,000 describing vectors obtained in the second phase of training and input them to a 1-NN classifier. As a result we obtain an indexed database containing the following information as shown in Fig. 4.

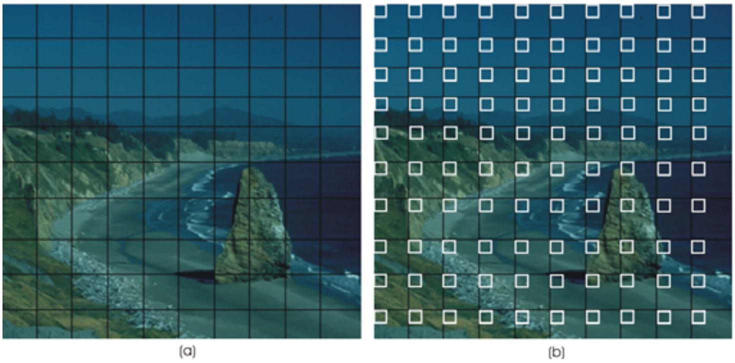


Fig. 3. (a) An image is uniformly divided into 100 sub-images to get 100 describing features. (b) For each sub-images, a window of 10×10 pixels is selected to compute the corresponding describing vector.

C1	C2	C3	C4	C5	C6	→	Name of Image
40	16	23	20	1	0	→	Image 1.jpg
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
28	19	9	9	15	20	→	Image k.jpg
							⋮
7	23	7	32	19	12	→	Image 300.jpg

Fig. 4. Form of the indexed database

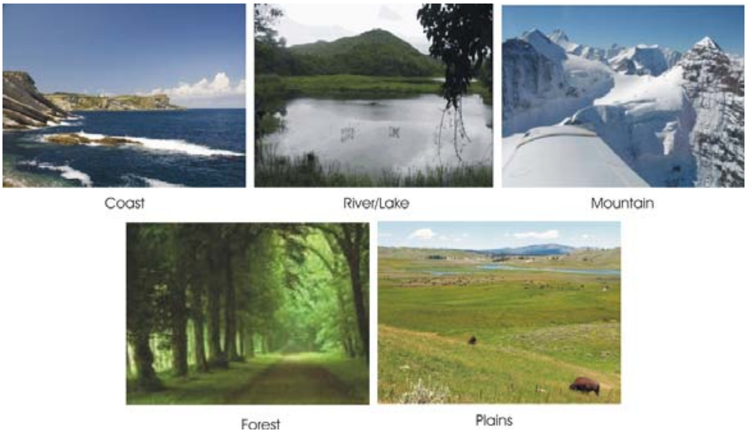


Fig. 5. Five different image classes have been manually chosen for image retrieval purposes

Retrieval stage. This stage is divided into the phases shown in Fig. 1(b). As shown, a query image is presented to the system. To this image the same feature extraction phases used during training are applied. As a result we get 100 describing vectors, These 100 vectors are presented to already trained 1-NN classifier. As a result we just get one indexing vector. This vector contains the probability that each one of the six classes C1, C2, C3, C4, C5 and C6 is contained in the query image. This vector is compared with the 300 vectors saved in the indexed database. To reduce the computing time and to get better retrieval results, we just take into account the two higher components of the six classes. As a distance we use the Euclidean distance. For retrieval purposes we have chosen five different kinds of images as show in Fig. 5. These five different types of images were manually selected.

Note. For testing our proposal we have chosen 300 natural images of the Corel Image Database (720×480). These images were provided by J. Vogel [4], [5], [6] and [15]. The 300 images used for training were grouped into the 5 types of images as follows: 54 mountains images, 54 lakes images, 54 coastal images, 54 forest images, 54 prairies images, and 30 clouds images.

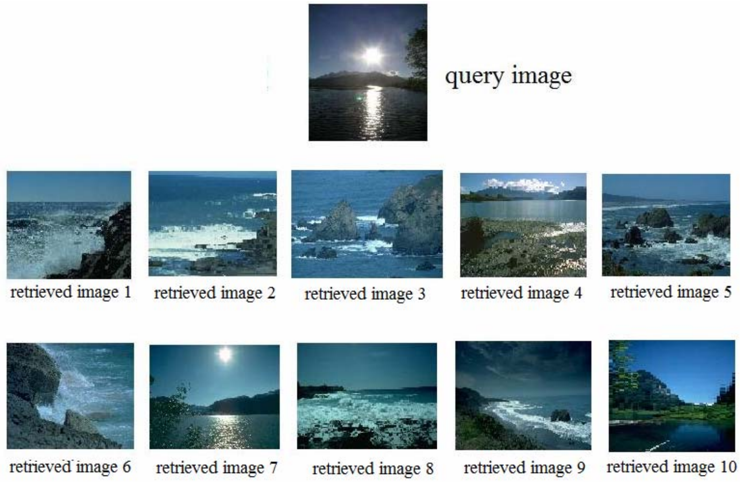


Fig. 6. Images retrieved given a query image of a sunset

3 Experimental Results

In this section we present the experimental results that validate our proposal. For this we have selected from Internet 221 images. These images are different from those used for training. We presented each of these 221 images to the system and asked it to show us the most 10 similar images from the indexed database. Figure 6 shows a query example. From Fig. 6 we can see for example that the system retrieves correctly 9 images and retrieves incorrectly 1 image (image 10). This gives a 90% of efficiency for this retrieval (full test can be shown in figures 7, 8 and 9). To test the efficiency of the proposal we have used the following two measures:

$$P = \frac{\text{No. of relevant images retrived}}{\text{Total no. of images retrieved}} \tag{1}$$

$$R = \frac{\text{No. of relevant images retrived}}{\text{Total no. of relevant images in database}} \qquad (2)$$

The first measure represents the number of relevant images retrieved with respect to the total number of images asked to be retrieved. The second measure represents the relevant images retrieved with respect to the total number of images used for training for a given class.

Fig. 7 shows the performance of our proposal against the method described in [14]. As we can appreciate our proposal performed a little better.

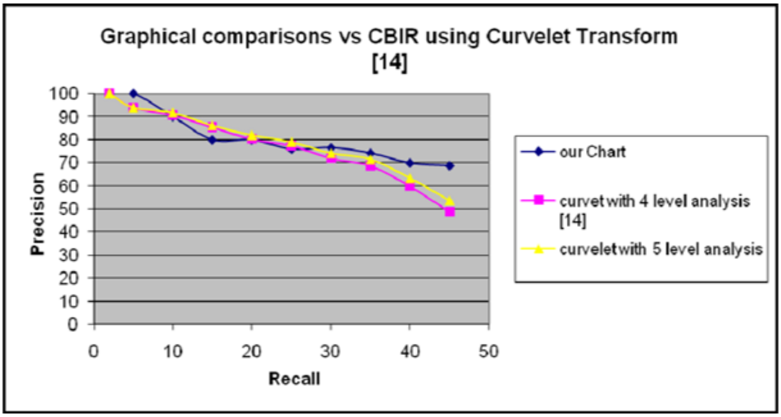

Graphical comparisons vs CBIR using Curvelet Transform [14]

Fig. 7. Performance of our proposal against the method described in [14]. We get a 79.05%, while in [14] they get a 77.71% of efficiency when using as a query the coastal image shown in Fig. 6.

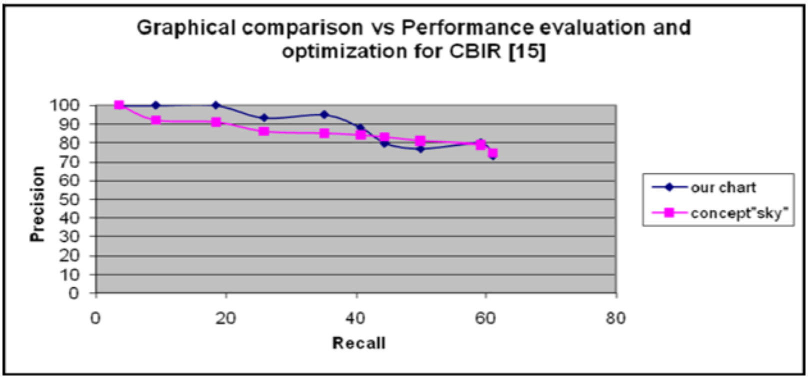

Graphical comparison vs Performance evaluation and optimization for CBIR [15]

Fig. 8. Performance of our proposal against the method described in [15]. We get a 85.93%, while in [15] they get a 85.61% of efficiency when using as a query a red sunset image.

In Figures 8 and 9, we compare our proposal against the method reported in [15]. As can be seen the performance of our proposal is just a little better than the one reported in [15].

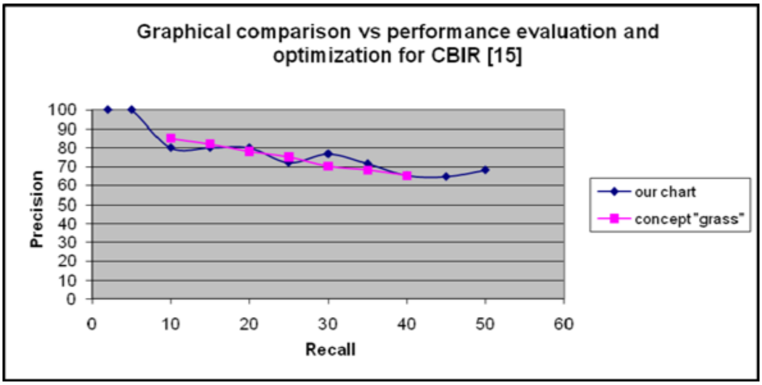


Fig. 9. Performance of our proposal against the method described in [15]. We get a 77.16%, while in [15] they get a 74.17% of efficiency when using as a query the forest image.

4 Conclusions

In this paper we have described a methodology that allows to automatically retrieving natural images from a database. During learning the proposal takes as input a set of images divided into five classes: coasts, lake/rivers, mountains, forests and plains. It extracts from them describing features from sets of points randomly and automatically selected. A *K*-means classifier is used to form six different clusters from the describing features obtained from the randomly and automatically chosen points. A 1-NN classifier is used to build an indexed database from the combination of all the describing vectors.

During retrieval the already trained 1-NN classifier is used to retrieve from the indexed database the most similar images given a query image. The experimental results show that our proposal performs better than two reported method in the literature. For this we have used the precision/recall measure.

Nowadays we are testing the proposal with more images and with more types of image classes and with more cluster regions. Also we are trying to use interest point detectors to select the points from which the describing vectors are going to be computed. We are also going to test with other describing features and other classifiers.

Acknowledgments. This work has been supported by the National Polytechnic Institute of Mexico (SIP-IPN), under grants 20090620 and 20091421, by the Mexican Science and Technology National Council (CONACyT), under grant 182938 and the European Commission and CONACYT under grant FONCICYT 93829.

References

[1] Long, F., Zhang, H.J., Feng, D.D.: Fundamentals of Content Image retrieval. In: Feng, D. (ed.) Multimedia Information Retrieval and Management. Springer, Heidelberg (2003)

[2] del Bimbo, A.: A Perspective View on Visual Information Retrieval Systems. In: Workshop on Content Based Access of Image and Video Libraries, vol. 21, pp. 108–109. IEEE, Los Alamitos (1998)

[3] Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image Retrieval: Ideas, Influences, and Trends of the New Age. ACM Computing Surveys 40(2), paper 5 (April 2008)

[4] Vogel, J., Schiele, B.: Semantic Modeling of natural scenes for content-based image retrieval. International Journal of Computer Vision 72(2), 133–157 (2007)

[5] Vogel, J., Schiele, B.: Semantic Modelling of Natural Scenes for Content-Based Image Retrieval. Int. J. of CV (2006), doi:10.1007/s11263-006-8614-1

[6] Vogel, J., Schwaninger, A., Wallraven, C., Bülthoff, H.H.: Categorization of natural scenes: local vs. global information. In: Proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV 2006), June 2006, pp. 33–40. ACM Press, New York (2006)

[7] Gonzalez Garcia, A.C.: Image retrieval based on the contents. PhD Thesis. Center for Research in Computing (CIC)-IPN, Mexico DF (September 2007)

[8] Presutti, M.: Co-currency Matrix in Multispectral Classification: Tutorial for Educators textural measures. In: The 4th day Educacao em Sensoriamento Remote Ambito not do Mercosul, Sao Leopoldo RS. Brazil, Augest 11-13 (2004)

[9] Rui, Y., Huang, Th.S., Chang, Sh.F.: Image Retrieval: Currente techniquies, Promissing Directions, and open Issues. Journal of Visual Communication and Image Representation 10, 39–62 (1999)

[10] Liu, Y., Zhang, D., et al.: A survey of Content-Based Image Retrieval with high-level semantics. Pattern Recognition 40, 262–282 (2007)

[11] Li, J., Wang, J.Z.: Real-Time Computerized Annotation of Pictures. In: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 911–920 (2006)

[12] Hiremath, P.S., Pujari, J.: Content Based Image Retrieval using Color, Texture and Shape features. In: 15th International Conference on Advanced Computing and Communications, pp. 780–784 (2007)

[13] Fukunaga, K.: Introduction to statistical Pattern Recognition. Academic Press, New York (1990)

[14] Sumana, I.J., Islam, M.M., Zhang, D., Lu., G.: Content Based Image Retrieval using curvelet transform. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing, October 8-10 2008, pp. 11–16 (2008)

[15] Vogel, J., Schiele, B.: Performance evaluation and optimization for Content-Based Image Retrieval. Pattern Recognition 39(5), 897–909 (2006)

Use of Ultrasound and Computer Vision for 3D Reconstruction

Ruben Machucho-Cadena[1], Eduardo Moya-Sánchez[1],
Sergio de la Cruz-Rodríguez[2], and Eduardo Bayro-Corrochano[1]

[1] CINVESTAV, Unidad Guadalajara, Departamento de Ingeniería Eléctrica y
Ciencias de la Computación, Jalisco, México
{rmachuch,emoya,edb}@gdl.cinvestav.mx
[2] Instituto Superior Politécnico "José A. Echeverría", Havana, Cuba
sergio@electrica.cujae.edu.cu

Abstract. The main result of this work is an approach for reconstructing the 3D shape and pose of tumors for applications in laparoscopy from stereo endoscopic ultrasound images using Conformal Geometric Algebra. We record simultaneously stereo endoscopic and ultrasonic images and then the 3D pose of the ultrasound probe is calculated using conformal geometric algebra. When the position in 3D of the ultrasound probe is calculated, we compound multiple 2D ultrasound images into a 3D volume. To segment 2D ultrasound images we have used morphological operators and compared its performance versus the obtained with segmentation using level set methods.

1 Introduction

Endoscopy became an increasing part of daily work in many subspecialties of medicine and the spectrum of applications and devices has grown exponentially [1]. The use of a stereoendoscope (i.e. an endoscope with two cameras instead of one) provides more information of the scenario, that is, two slightly different views of the same scene at the same time allows the calculation of the spatial coordinates [2]. On the other hand, the ultrasound is found to be a rapid, effective, radiation free, portable and safe imaging modality [3]. However, the endoscopic images can not see beyond opaque or occluded structures. The incorporation of ultrasound images into stereoendoscope operative procedures generating more visibility in the occluded regions.

By superimposing Stereo Endoscopic Ultrasound (SEUS) images in the operative field (in the case of laparoscopic or robotic procedures), it would be possible integrate them into a 3D model. This 3D model can help surgeons to better locate some structures such as tumors during the course of the operation. Virtually all available methods use either a magnetic or optic tracking system (and even a combination of both) to locate the tip of the US probe (USP) in 3D space[4]. These systems are sometimes difficult to implement in intraoperative scenarios, because the ferromagnetic properties of surgical instruments can affect magnetic tracking systems [5].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 782–789, 2009.

We extent a previously proposed method [4] which used just monocular endoscopic images to calculate the pose (i.e. the position and orientation) of the USP. In this paper we use stereo endoscopic images and apply our approach in laparoscopy.

2 System Setup

Our experimental setup is illustrated in Figure 1. The equipment setup is as follows: The endoneurosonography equipment (ENS) provides an ultrasound probe (USP) that is connected to an ultrasound system Aloka. The USP is introduced through a channel in the stereo endoscope (Fig. 1a) and is observed by the endoscope.

The USP is flexible and is rotating around the longitudinal axis at about 60 rpm. It can also move back and forth and since the channel is wider than the USP there is also a random movement around the channel. The US image is orthogonal to the USP axis. We know that in small interval of time Δt, the USP is fixed, and the two endoscopic cameras undergo a movement, which is equivalent to an inverse motion, that is, the endoscopic camera is fixed, and ultrasound probe moves.

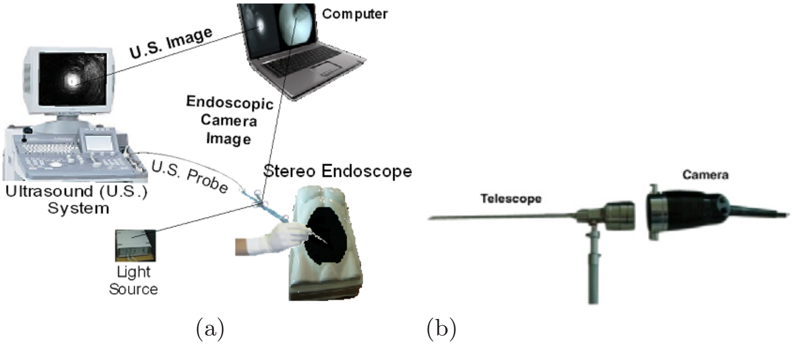
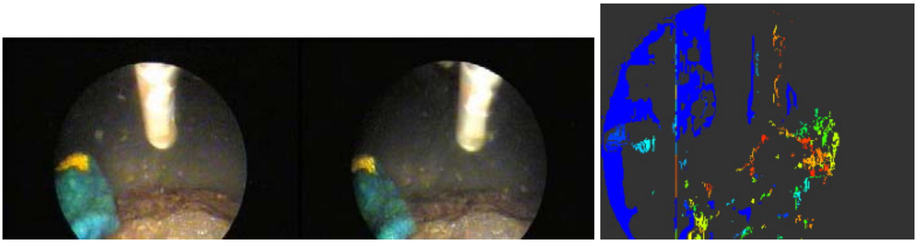


Fig. 1. a) Experimental setup. b) Equipment, telescope and camera of the stereo endoscope.

3 Tracking the Ultrasound Probe

We have used multiple view geometry to process the stereo images; the Figure 2a shows a pair of rectificated stereo images and Fig. 2b is its depth map. The cameras were calibrated using the method described in [6]. We track the USP throughout the endoscopic camera images.

In order to track the USP we have used the particle filter and an auxiliary method based on thresholding in the HSV-Space in order to improve the tracking as follows:



(a) (b)

Fig. 2. a) Pair of rectificated stereo images. b)its depth map.

3.1 The Particle Filter Tracker

We used the particle filter to found the USP in the endoscopic images in a similar way to our previous paper [4], with addition of some heuristics. The resultant process is:

- Throw 35 particles on each image and test the likelihood for each of them with respect to a reference histogram. The likelihood is the distance between the color histograms of the model and the current particle. The particle which have the higher likelihood value will be the winner. Then we take the orientation, position and scale from the particle winner as an estimation of the USP in the endoscopic camera image. If the likelihood value of the best particle is higher than a threshold value, we use this information to update the reference model, that is the reference histogram. We have used a threshold value of 0.17 to update the reference model.
- If the likelihood value of the best particle is lower than a threshold value, then we set a flag to enable an additional tracking process in order to find the USP; this additional process is explained in section 3.2. We enable the additional tracking process when the likelihood value is less than 0.08, otherwise this additional process is disabled.
- We select the 35 particles from a set of particles that we have built for the possible positions, orientations and scales of the USP in the endoscopic image. This set of particles is built off-line and is saved in a text file by using only a chain code for each particle. We also save the size (scale) and the orientation for each particle. In the beginning of the tracking process we read the text file just one time and we store this information in a data structure.

The threshold values and the number of particles (35) were experimentally obtained. Figure 3a shows the model used to obtain the initial reference histogram. Figure 3b shows a mask used to build the set of particles. We can see here four points with different colors used as reference in the construction process of the set of particles. The position of the four points is also saved in the text file afore-mentioned are used to identify the top of the particle in order to place it correctly on the frame of the endoscopic camera. Figure 3c shows the used camera frame. It is built just one time at beginning of the tracking process and Figure 4 shows some particles taken from the set of particles. The results of the tracking of the USP are show in Figure 6 we have an example of tracking of the ultrasound probe in the stereo endoscopic camera by using particle filter.

3.2 Tracking Based on Thresholding in the HSV-Space

The tracking process is based on thresholding in the HSV-Space and it is used as an auxiliary to find the USP in the endoscopic camera when the likelihood value of the best particle is lower than a threshold value. This works as follows:

- Make a copy of the original image (to preserve it).
- Convert the original image from RGB to HSV-Space.
- Make a binary image by selecting an interval of values from the saturation histogram. This interval should separate the USP from the rest of the image. We have selected the interval [0, 50] where we observe the first peak of the histogram.

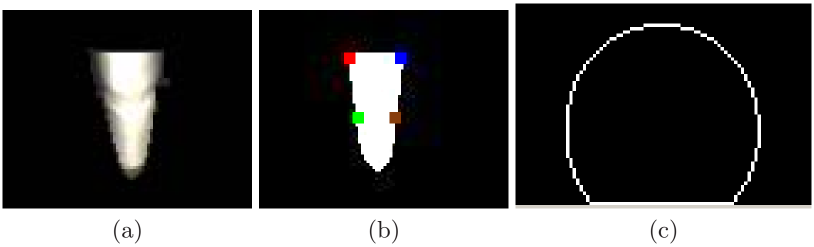


Fig. 3. a)Model used to obtain the initial reference histogram. b) Model mask used to build the set of particles. c) Camera frame, it is built just one time at the beginning of the tracking process.

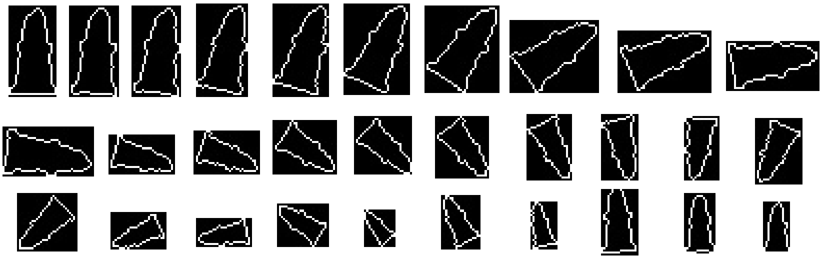


Fig. 4. Set of particles, some particles selected

Build a new image called sectors, from the binary image and the Canny filter as follows:

- If the pixel (x, y) has an edge (from Canny) and it is part of the mask (binary image) then (x, y) will belong to sectors, see Fig. 5e.
- Apply the closing morphological operator to the image sectors in order to fill small holes and to join broken edges. See Fig. 5f.
- Apply the chain code to calculate the smallest areas of the image sectors, and eliminate them. see Fig. 5g.
- Get the initial, middle and final of the segmented USP (from the image sectors) on the camera frame and use this information to throw nine particles (replacing the nine particles with the lower likelihood values) in order to improve the tracking process.

Figure 5a illustrates the saturation histogram for the endoscopic camera image shown in Fig. 5b. Figure 5c shows its saturation image and Figure 5d is the application of the Canny filter to the original image. This tracking method is just used as a support method because it does not take into account temporal information and because is also sensible to partial occlusions of the USP in the endoscopic cameras images as well as background, illumination and contrast variations.

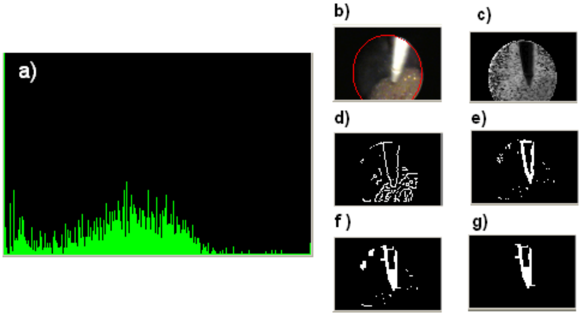


Fig. 5. Tracking process in the HSV-Space

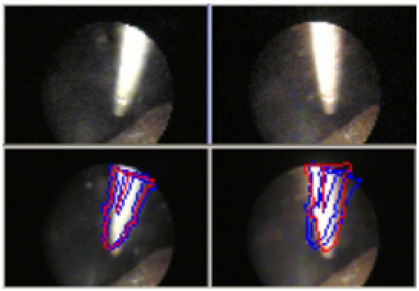


Fig. 6. Example of Tracking of the USP. The best particle is shown in red color.

4 Ultrasound Image Processing

We have used two methods in order to process the ultrasound images; the first one is based in morphological operators [4] and the second one is the level sets method.

The level sets method uses an initial seed on the image. This seed evolves with the time until a zero velocity is reached or the curve is collapsed (or a maximum number of iterations is reached). To evolve the curve, the method uses two lists called `Lin` and `Lout` [7]. We present the results of the processing and a comparison between both methods. They work independently of the tumor characteristics.

Figure 7 shows the results obtained by using morphological operators. Figure 7a is an original ultrasound image. In Figure 7b the central part of the image is excluded, because it only contains noise and the ROI is selected. The binary mask obtained for this method that will be applied to the original image is showed in Figure 7c and Figure 7d shows the result of the segmentation.

Figure 8 shows the results obtained by using the level sets method. Figure 8a is an original ultrasound image. Figure 8c shows the ROI selected. Figure 8d shows the initial seed applied to Figure 8c. Figure 8e shows the collapsed curve. Figure 8f is the binary mask obtained from Figure 8e. This mask is applied to the original image and so we have obtained the result of the segmentation (Figure 8b. Both figures were obtained from the Aloka ultrasound equipment by using a box shaped rubber phantom.

4.1 Comparison between Both Methods

We have obtained a processing time of 0.005305 seconds for the morphological operators method versus 0.009 seconds for the level sets method, that is 188 fps versus 111 fps. We recommend both methods for in line implementation, because they are fast and reliable.

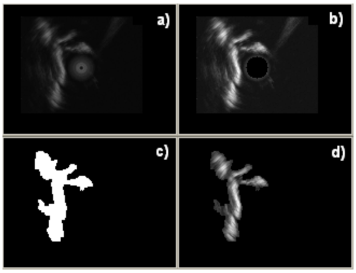


Fig. 7. Isolating the tumor. a) Original US image to be segmented. b) The central part of the image is excluded. c) ROI. d) Result of segmentation.

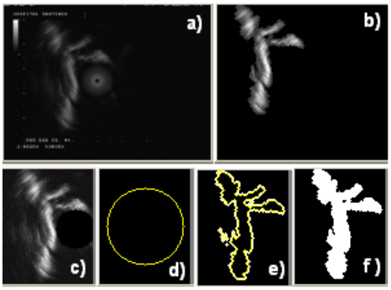


Fig. 8. Segmentation using the level sets method

5 Calculating the 3D Pose of the Tumor

This work make use Conformal Geometric Algebra (CGA) to represent geometric entities (points, lines, planes, spheres, etc.) in a compact and powerful form [8]. The CGA preserves the Euclidean metric and adds two basis vectors: e_+, e_- (where ${\mathsf{e}_+}^2 = 1$ and ${\mathsf{e}_-}^2 = -1$), which are used to define the point at the origin $\mathsf{e}_0 = \frac{1}{2}(\mathsf{e}_- - \mathsf{e}_+)$ and the point at the infinite $\mathsf{e} = \mathsf{e}_- + \mathsf{e}_+$. The points in CGA are related with the Euclidean space by: $\underline{p} = \mathbf{p} + \frac{\mathbf{p}^2}{2}\mathsf{e} + \mathsf{e}_0$. A sphere in dual form is represented as the *wedge* of four conformal points that lies on sphere $\underline{s}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \underline{d}$, its radius ρ and its center $\underline{p}$ in $\mathcal{R}^3$ can be obtained using: $\rho^2 = \frac{\underline{s}^2}{(\underline{s} \cdot \mathsf{e})^2}$, $\underline{p} = \frac{\underline{s}}{-(\underline{s} \cdot \mathsf{e})} + \frac{1}{2}\rho^2\mathsf{e}$. A plane in dual form is defined as a sphere, but the last point is at the infinity: $\underline{\pi}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \mathsf{e}$. A line in dual form is represented as the *wedge* of two points and the infinity point: $\underline{L}^* = \underline{a} \wedge \underline{b} \wedge \mathsf{e}$. A line can also be calculated as the intersection of two planes: $\underline{L} = \underline{\pi_1} \wedge \underline{\pi_2}$. This equation is used to calculate the 3D line that represents the ultrasound probe axis. To achieve a translation by a distance d_2 from a point $\underline{p_1}$ in the direction of a line and to obtain $\underline{p_2} : T = exp\left(\frac{1}{2}d_2L\right)$, $\underline{p_2} = T\underline{p_1}\widetilde{T}$. The last equation is used to find the position of the ultrasound sensor in order to put the segmented ultrasound image in 3D space; where $\underline{p_1}$, $\underline{p_2}$ represent the begin and the end respectively of the best particle on the stereo endoscopic images and d_2 is the retroprojected distance between them.

5.1 Results

Figure 9a shows a convex hull applied to a set of slices of tumor in the 3D space and, Figure 9b shows the part of the phantom used and the tumor.

6 Conclusions

We have addressed the problem of obtaining 3D information from joint stereo endoscopic and ultrasound images obtained with SEUS equipment. We have used

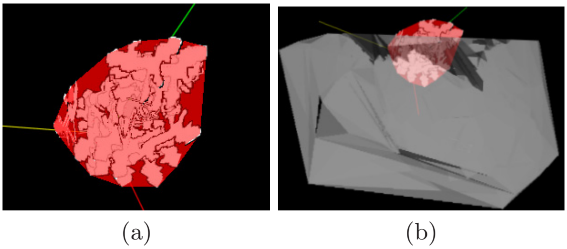


Fig. 9. 3D Reconstruction from a set of slices of tumor

conformal geometric algebra to calculate the pose of the ultrasound sensor in order to put the segmented tumor in the 3D space. To verify the segmentation of the tumor in the ultrasound images, we have compared two different segmentation methods and obtained good results at a reasonable speed for both. Some preliminary results are presented.

References

1. Muller, U.: Possibilities and limitations of current stereo-endoscopy. Surg. Endosc 18, 942–947 (2004)
2. Senemaud, J.: 3D-Tracking of minimal invasive instruments with a stereoendoscope. PhD thesis, Universität Karlsruhe, Karlsruhe, Germany (2008)
3. Nascimentoa, R.G., Jonathan, C., Solomon, S.B.: Current and future imaging for urologic interventions. Current Opinion in Urology 18, 116–121 (2008)
4. Machucho Cadena, R., de la Cruz Rodriguez, S., Bayro Corrochano, E.: Joint freehand ultrasound and endoscopic reconstruction of brain tumors, pp. 691–698 (2008)
5. Hummel, J.B., et al.: Design and application of an assessment protocol for electromagnetic tracking systems. Medical Physics 32, 2371–2379 (2005)
6. Bayro Corrochano, E., Daniilidis, K.: The dual quaternion approach to hand-eye calibration. In: International Conference on Pattern Recognition, vol. 1 (1996)
7. Shi, Y.: Object-based Dynamic Imaging with Level Set Methods. PhD thesis, Boston University, Boston, U.S.A. (2005)
8. Bayro Corrochano, E.: Geometric Computing for Perception Action Systems: Concepts, Algorithms, and Scientific Applications. Springer, Heidelberg (2001)

Two-Frame Optical Flow Formulation in an Unwarping Multiresolution Scheme

C. Cassisa[1,2], S. Simoens[1], and V. Prinet[2]

[1] Lab. of Fluid Mechanics and Acoustics (LMFA),
Ecole Centrale Lyon, France
[2] Lab. of Informatics, Automatics and Applied Mathematics (LIAMA),
Chinese Academy of Sciences, Beijing, Chine
ccassisa@ec-lyon.fr

Abstract. In this paper, we propose a new formulation of the Differential Optical Flow Equation (DOFE) between two consecutive images considering spatial and temporal information from both. The displacement field is computed in a Markov Random Field (MRF) framework. The solution is done by minimization of the Gibbs energy using a Direct Descent Energy (DDE) algorithm. A hybrid multiresolution approach, combining pyramidal decomposition and two-step multigrid techniques, is used to estimate small and large displacements. A new pyramidal decomposition method without warping process between pyramid levels is introduced. The experiments carried out on benchmark dataset sequences show the effectiveness of the new optical flow formulation using the proposed unwarped pyramid decomposition schema.

Keywords: Optical flow estimation, RMF minimization, Multiresolution technique.

1 Introduction

Motion estimation has always been a major activity in computer vision community, with application in tracking, stereo matching, rigid and elastic motions, fluid propagation... Since early 80's, it has been well studied and many approaches have been proposed. But it is still remaining challenging to this date. For details on existing algorithms, you can refer to Barron's et al. [4].

Differential Optical Flow Equation (DOFE), introduced by Horn & Schunck [9], has proved to be very powerful in motion estimation. The DOFE is based on the hypothesis of illumination constancy over a small period of time. At the beginning, approaches were defined on a centered formulation of the DOFE that needs at least three successive images ([9,4]). Other approaches studied the case of only two successive frames and proposed a non-centered DOFE based on the first image ([5]) or on the second one ([11,13]). Recently the work of Alvarez et al. [1] imagines an intermediate image at the half way from the first to the second image and uses a symmetrical formulation of DOFE based on two images. However, this method needs many interpolation and warping steps that can affect the quality of the estimation.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 790–797, 2009.

The hypothesis of small displacement made to define DOFE is very restrictive. Most of movements do not respect this assumption over image domain. To deal with it, multiresolution techniques are commonly used. The idea is to estimate the displacement field in an incremental and iterative way for different image resolution (coarse-to-fine). The different resolutions can be generated by scale-space theory ([2]) that convolves Gaussian filter of different variances to the image to extract coarse to fine information or by pyramidal decomposition of original images into successive images of smaller resolution. Many decomposition techniques were proposed as Gaussian pyramid [5,11,13] , steerable pyramid [15] or wavelet decomposition [10]. But all these approaches, during the multiresolution process, warp the image by the coarse displacement field estimated at upper pyramid level before computing the missing incremental displacement between the warped and the other image. This step transforms and interpolates image information. It is strongly correlated to the quality of coarse displacement field.

These last years, many works has been done on the search of the optimal solution of the displacement field. Due to the non-convexity of problem formulation, multigrid technique is often used ([12]). It allows local minimization to not be trapped in local minima. It has been shown that coupling multiresolution and multigrid techniques for optical flow estimation can improve the accuracy of the estimation ([8,7]).

In the present work, we propose a new non-centered formulation of the DOFE that refers to the two image spatial information (TI_DOFE). The DOFE is solved by maximizing a posterior probability using a Direct Descent Energy (DDE) algorithm through the minimization of an equivalent MRF-Gibbs energy. Making a local spatial assumption, we define a multiresolution technique that does not need to warp image between two pyramid levels. As the previous work in [7], the multiresolution is combined with a two-step multigrid technique helping DDE to converge to the optimal solution while improving significantly the computational time.

The rest of the paper is organized as follows. Section 2 defines TI_DOFE, formulates the MRF framework and introduces the minimization method. In section 3, we detail the pyramidal multiresolution schema using warping or unwarping steps and the combined multigrid technique. Results about three different sequences are illustrated and discussed in section 4. Section 5 concludes the paper.

2 Methodology

2.1 Two-Frame Optical Flow Equation

For a two-frame temporal image sequence, the optical flow equation (OFE) is the 2D vector field of apparent displacement $\mathbf{d}(\mathbf{s}) = (dx(\mathbf{s}), dy(\mathbf{s}))$ that links pixels $\mathbf{s} = (x, y)$ of the first image at time t with its correspondent position in the second image at time $t + \Delta t$.

OFE definition is based on the assumption that the image illumination $(I(\mathbf{s}, t))$ is constant over a small time interval Δt:

$$I(\mathbf{s}+\mathbf{d}(\mathbf{s}),t+\varDelta t)-I(\mathbf{s},t)\approx 0 \tag{1}$$

Making the hypothesis of a small displacement $\mathbf{d}(\mathbf{s})$ over a small time interval $\varDelta t$, the Differential Optical Flow Equation (DOFE) can be computed from a 1^{st} order Taylor expansion of $I(\mathbf{s}+\mathbf{d}(\mathbf{s}),t+\varDelta t)$ around $\mathbf{s}$:

$$I(\mathbf{s}+\mathbf{d}(\mathbf{s}),t+\varDelta t)=I(\mathbf{s},t+\varDelta t)+\mathbf{d}(\mathbf{s}).\nabla I(\mathbf{s},t+\varDelta t)+\vartheta(\mathbf{d}^2(\mathbf{s})) \tag{2}$$

If $\mathbf{d}(\mathbf{s})$ is small enough, $\vartheta(\mathbf{d}^2(\mathbf{s}))$ can be neglected. We have the following DOFE:

$$I(\mathbf{s},t+\varDelta t)-I(\mathbf{s},t)+\mathbf{d}(\mathbf{s}).\nabla I(\mathbf{s},t+\varDelta t)\approx 0 \tag{3}$$

With $\nabla I(\mathbf{s},t+\varDelta t)=(\frac{\partial I(x,y,t+\varDelta t)}{\partial x},\frac{\partial I(x,y,t+\varDelta t)}{\partial y})$ spatial gradients at time $t+\varDelta t$ (second image). We call this equation DOFE_2.

Doing 1^{st} order Taylor expansion of $I(\mathbf{s}+\mathbf{d}(\mathbf{s}),t+\varDelta t)$ around $\mathbf{s}$ and $\varDelta t$. For $\mathbf{d}(\mathbf{s})$ and $\varDelta t$ small enough, $\vartheta(\mathbf{d}^2(\mathbf{s}),\varDelta t^2)\approx 0$ and DOFE can be rewrite as:

$$\varDelta t.I_t(\mathbf{s},t)+\mathbf{d}(\mathbf{s}).\nabla I(\mathbf{s},t)\approx 0 \tag{4}$$

With $I_t(\mathbf{s},t)=\frac{\partial I(x,y,t)}{\partial t}$ and $\nabla I(\mathbf{s},t)=(\frac{\partial I(x,y,t)}{\partial x},\frac{\partial I(x,y,t)}{\partial y})$ the temporal and spatial gradients at time t (first image). Let call it DOFE_1.

The finite difference of the temporal gradient $I_t(\mathbf{s},t)$ using the two-frame image sequence is:

$$I_t(\mathbf{s},t)=\frac{I(\mathbf{s},t+\varDelta t)-I(\mathbf{s},t)}{\varDelta t} \tag{5}$$

From eq.4 and eq.3, We obtain then a new non-centered DOFE that contains spatial information from both images. We call it TI_DOFE:

$$\varDelta t.I_t(\mathbf{s},t)+\mathbf{d}(\mathbf{s}).\frac{1}{2}\left(\nabla I(\mathbf{s},t)+\nabla I(\mathbf{s},t+\varDelta t)\right)\approx 0 \tag{6}$$

2.2 MRF Framework and Minimization

The displacement field $\mathbf{d}$ is considered as a random variable that maximizes a joint probability. It is computed within a MRF framework via Maximum a Posteriori estimation using a Bayesian decomposition of a Gibbs distribution.

$$P(\mathbf{d}(\mathbf{s}),I(\mathbf{s}))=\frac{1}{Z}e^{-E(\mathbf{d}(\mathbf{s}),I(\mathbf{s}))} \tag{7}$$

Where Z is the normalization constant and the total Gibbs energy E is defined by:

$$E(\mathbf{d}(\mathbf{s}),I(\mathbf{s}))=\sum_{\mathbf{s}\in C_1}V_d(\mathbf{d}(\mathbf{s}),I(\mathbf{s}))+\sum_{\mathbf{s},\mathbf{s}'\in C_2}\alpha_p\,V_p(\mathbf{d}(\mathbf{s}),\mathbf{d}(\mathbf{s}')) \tag{8}$$

$I(\mathbf{s})$ represents the observed data extracted from image intensities. C_1 and C_2 are respectively the single-site and pair-site cliques. α_p is a weighting coefficient that is used to play on the influence of the data term V_d compared to the prior term V_p. V_p only depends of its 4-neighborhood ($\mathbf{s}'$ neighbor of $\mathbf{s}$).

Data term is a quadratic function of DOFE_1 (eq.4), DOFE_2 (eq.3) or TI_DOFE (eq.6). Eq.9 represents the case of TI_DOFE:

$$Vd(\mathbf{d}(\mathbf{s}), I(\mathbf{s})) = \left(\varDelta t.I_t(\mathbf{s}, t) + \mathbf{d}(\mathbf{s}).\frac{1}{2} \left(\nabla I(\mathbf{s}, t) + \nabla I(\mathbf{s}, t + \varDelta t)\right) \right)^2 \tag{9}$$

DOFE does not admit a unique solution. To solve the ill-posed problem, we add a prior term (regularization) that reduces the configuration of possible solutions. The prior term is defined as Tikhonov regularization [16]:

$$Vp(\mathbf{d}(\mathbf{s}), \mathbf{d}(\mathbf{s}')) = ||\mathbf{d}(\mathbf{s}) - \mathbf{d}(\mathbf{s}')||^2 \tag{10}$$

The minimization of the energy is achieved by a Direct Descent Energy (DDE). DDE consists to minimize $E(\mathbf{d}(\mathbf{s}), I(\mathbf{s}))$ by successive iterations over all pixels. A small incremental $\delta\mathbf{d}(\mathbf{s})$ random value is generated where $\mathbf{d}(\mathbf{s}) \leftarrow \mathbf{d}(\mathbf{s}) + \delta\mathbf{d}(\mathbf{s})$. $\delta\mathbf{d}(\mathbf{s})$ is conserved only if $E(\mathbf{d}(\mathbf{s}), I(\mathbf{s}))$ is decreased. This minimization method converges to a local minimum of the energy. It is then dependent to the initialization of displacement field. To cope with this problem, the weighting coefficient $\alpha_p(i)$ is logarithmic increasing over the iteration (i) from 0 to α_p. In this way, the estimated displacement field satisfies first the DOFE then it is slowly becoming more constrained by the regularization term. Moreover, the multigrid technique allows the DDE minimization to not be trapped into local minima and to reach an optimal solution.

3 Combined Multiresolution - Multigrid

3.1 Pyramidal Decomposition

The multiresolution by pyramidal decomposition from coarse to fine resolution has been proved to be numerically useful for optical flow estimation [14]. The image resolution is iteratively reduced in a pyramid of K different successive resolution levels from the original resolution using Gaussian filter [6]. We use a Gaussian filter of variance $\sigma = 1$.

At each pyramid level k, the total displacement field $\mathbf{d}^k = \tilde{\mathbf{d}}^{k+1} + \mathbf{d'}^k$ where $\tilde{\mathbf{d}}^{k+1}$ is the interpolated total displacement field computed at coarser resolution $(k+1)$ and $\mathbf{d'}^k$ is the complementary displacement field at level k. $\mathbf{d'}^k$ is small at each pyramid level k. The 1^{st} order Taylor expansion condition is then respected for each level. The TI_DOFE becomes:

$$\varDelta t.I_t(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t) + \mathbf{d'}^k.\frac{1}{2} \left(\nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t) + \nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \varDelta t)\right) \approx 0 \tag{11}$$

For better readability, we did not write the spatial dependency of the displacement field ($\tilde{\mathbf{d}}^{k+1}(\mathbf{s})$, $\mathbf{d'}^k(\mathbf{s})$). By similarity, DOFE_1 and DOFE_2 for multiresolution can easily be obtained in the same way.

To compute the observed data of the equation, common methods warp the image (I) into a compensated intermediate image $(\hat{I})$ depending to the used DOFE formulation : [5] warps the first frame to the second $\hat{I}(\mathbf{s}, t) = I(\mathbf{s}+\tilde{\mathbf{d}}^{k+1}, t)$

or [8,13] warps the second to the first $\hat{I}(\mathbf{s}, t+\Delta t) = I(\mathbf{s}+\tilde{\mathbf{d}}^{k+1}, t+\Delta t)$. The image (I) is transformed by the displacement field $\tilde{\mathbf{d}}^{k+1}$, then the intensity distribution is interpolated to a regular pixel grid $\mathbf{s}$. Notes that $\mathbf{s}$ in $\hat{I}(\mathbf{s}, t)$ represents in fact the old position $\mathbf{s} + \tilde{\mathbf{d}}^{k+1}$ and $\mathbf{s}$ in $\hat{I}(\mathbf{s}, t + \Delta t)$ still represents the same $\mathbf{s}$.

The multiresolution TI_DOFE with warping (W_MR) can be written as:

$$\Delta t.\hat{I}_t(\mathbf{s}, t) + \mathbf{d}'^k.\frac{1}{2}\left(\nabla\hat{I}(\mathbf{s}, t) + \nabla\hat{I}(\mathbf{s}, t + \Delta t)\right) \approx 0 \qquad (12)$$

The spatial and temporal gradients are computed from the warped images. Their precision depends to the quality of $\tilde{\mathbf{d}}^{k+1}$ estimation and to the efficiency of warping technique. $\hat{I}_t(\mathbf{s}, t) = I(\mathbf{s}, t + \Delta t) - \hat{I}(\mathbf{s}, t) = \hat{I}(\mathbf{s}, t + \Delta t) - I(\mathbf{s}, t)$.

In this paper, we proposed to suppress the warping step. We consider that the spatial derivatives are locally invariant over a small time interval Δt. It means that the gradients of DOFE can be computed on both original images. No image needs to be warped. Then we use the correct gradient quantity in respect to the coarse interpolated displacement field $\tilde{\mathbf{d}}^{k+1}(\mathbf{s})$ for each pixel $\mathbf{s}$.

The multiresolution TI_DOFE without warping (noW_MR) take the following form:

$$\Delta t.I_t(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t) + \mathbf{d}'^k.\frac{1}{2}\left(\nabla I(\mathbf{s}, t) + \nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \Delta t)\right) \approx 0 \qquad (13)$$

Where $\nabla I(\mathbf{s}, t) = \nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t)$ is the spatial gradient on the first image, $\nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \Delta t)$ is the spatial gradient on the second image at coordinates $\mathbf{s} + \tilde{\mathbf{d}}^{k+1}$, $\Delta t.I_t(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t) = I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \Delta t) - I(\mathbf{s}, t)$ is the difference of the intensity at coordinates $\mathbf{s} + \tilde{\mathbf{d}}^{k+1}$ on the second image with the intensity at $\mathbf{s}$ on the first one. We use a bilinear interpolation to compute the corresponding value of $\nabla I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \Delta t)$ and $I(\mathbf{s} + \tilde{\mathbf{d}}^{k+1}, t + \Delta t)$.

To resume, W_MR warps image information then compute spatial and temporal gradients. In noW_MR, due to local invariant hypothesis on gradients, gradients can be first computed on original images then only values in $\mathbf{s} + \tilde{\mathbf{d}}^{k+1}$ are interpolated.

The prior term along the multiresolution scheme is still the Tikhonov regularization of the total displacement field $\mathbf{d}^k(\mathbf{s})$:

$$Vp(\mathbf{d}^k(\mathbf{s}), \mathbf{d}^k(\mathbf{s}')) = ||(\tilde{\mathbf{d}}^{k+1}(\mathbf{s}) + \mathbf{d}'^k(\mathbf{s})) - (\tilde{\mathbf{d}}^{k+1}(\mathbf{s}') + \mathbf{d}'^k(\mathbf{s}'))||^2 \qquad (14)$$

3.2 Multigrid Method

At each pyramid level k, we use a two-step multigrid method previously proposed in [7]. The complementary displacement field $\mathbf{d}'^k$ is decomposed into a global component $\mathbf{d_g}'^k$ (average over a mesh size) and a local component $\mathbf{d_l}'^k$ (local deviation from $\mathbf{d_g}'^k$ for each pixel). $\mathbf{d_g}'^k$ is very fast to compute and furnishes a good approximation of the final displacement. It is used to initialize the search of $\mathbf{d}'^k = \mathbf{d_g}'^k + \mathbf{d_l}'^k$ at pixel level. Computational time is faster and the minimization can reach a better solution closer to the optimal.

4 Results

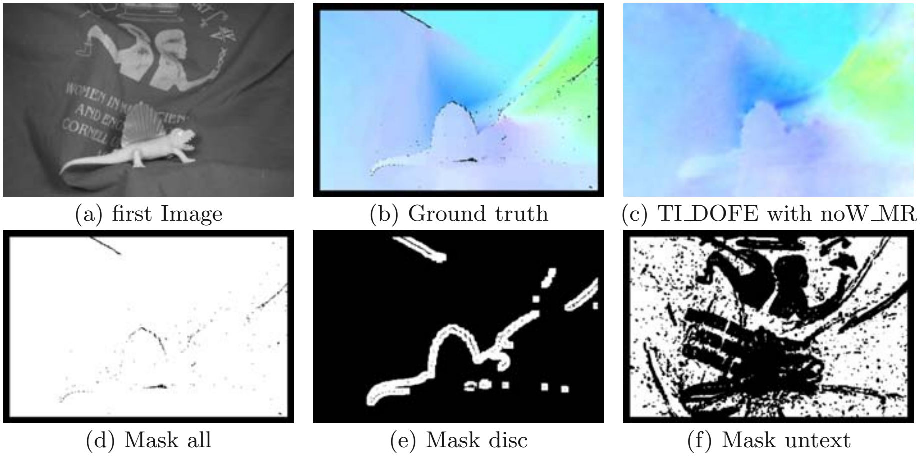
For all the illustrated results, we use the same parameter definitions to be able to compare the DOFE formulations and multiresolution schema. We set a 4-level pyramid decomposition for multiresolution and a grid size of 4×4 pixels for the multigrid method. The weighting coefficient α_p is spatially constant and equal to 100.

We evaluate the performance of our methods on the recent Middlebury optical flow benchmark dataset [3]. We use, in this paper, the Average Angle Error (AAE) [4,3] criteria to compare the efficiency of estimations:

$$AE = arccos\left(\frac{\mathbf{d_c}(\mathbf{s}).\mathbf{d_e}(\mathbf{s})}{||\mathbf{d_c}(\mathbf{s})||\ ||\mathbf{d_e}(\mathbf{s})||}\right) \tag{15}$$

AE is the angle error between the correct displacement $\mathbf{d_c}$ and the estimated displacement $\mathbf{d_e}$. The AAE is computed for three kinds of image area: all the image domain without border (all), the motion discontinuities (disc) and textureless regions (untext). We pre-process the data by convolving each frame of the sequence with a smoothing Gaussian filter ($\sigma = 1$).

We discuss the results of DOFE_1, DOFE_2 and TI_DOFE using W_MR or noW_MR on the dimetrodon sequence. Fig.1 shows the first input image, the ground truth where displacement vectors are coded with the color map proposed in [3] and the estimated displacement vector field computed using the two-frame optical flow formulation (TI_DOFE) with unwarping multiresolution scheme (noW_MR). The second line of Fig.1 illustrates the three masks used to compute AAE for all image domain, motion discontinuity area and textureless regions.


(a) first Image (b) Ground truth (c) TI_DOFE with noW_MR

(d) Mask all (e) Mask disc (f) Mask untext

Fig. 1. Dimetrodon: One of the three types of data illustrated in this paper. Ground truth and estimated field (TI_DOFE with noW_MR) are represented by flow field color coding map ([3]). Only white area is used to compute the AAE for the different masks.

The estimated field is very similar to the ground truth field. Because it is difficult to visualize the difference between the approaches, table 1 shows the different AAE values for three kinds of sequences (Dimetrodon, Yosemite and Venus) for the different DOFE formulation using warping and unwarping multiresolution. Performance of our methods is also compared to results shown in [3] of few classic optical flow algorithms.

Table 1. AAE comparison of our methods with classic algorithms ([3]) for Dimetrodon, Yosemite and Venus sequences. In bold: smallest AAE for classic algorithms, W_MR and noW_MR. In red: smallest AAE over all methods.

AAE	dimetrodon			yosemite			venus		
	all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	10.99	**9.41**	14.22	**1.69**	**2.86**	**1.05**	8.73	31.46	8.15
Black and Anandan	**9.26**	10.11	**12.08**	2.65	4.18	1.88	**7.64**	**30.13**	**7.31**
Pyramid LK	10.27	9.71	13.63	5.22	6.64	4.29	14.61	36.18	24.67
MediaPlayer TM	15.82	26.42	16.96	11.09	17.16	10.66	15.48	43.56	15.09
Zitnick et al.	30.10	34.27	31.58	18.50	28.00	9.41	11.42	31.46	11.12
W_MR DOFE_1	5.20	8.62	6.17	3.21	4.88	**1.33**	8.56	34.85	8.21
W_MR DOFE_2	5.43	8.72	6.19	3.49	**4.75**	2.01	9.57	35.17	9.02
W_MR TI_DOFE	**5.00**	**8.43**	**5.89**	**3.17**	4.81	1.35	**8.32**	**34.81**	**7.90**
noW_MR DOFE_1	5.12	8.50	6.02	2.89	**4.13**	1.23	9.03	35.28	8.71
noW_MR DOFE_2	4.99	**8.09**	5.80	2.93	4.15	1.12	8.72	34.37	8.72
noW_MR TI_DOFE	**4.92**	8.21	**5.80**	**2.88**	**4.13**	**1.06**	**8.41**	**33.81**	**8.54**

From the table, we can remark that our optical flow approach outperforms algorithms as Pyramid LK, MediaPlayer TM and Zitnick and that it gets around the same magnitude of AAE than Bruhn et al. and Black and Anandan. In bold red are the smallest AAE over all methods for each sequence and each method produces at least one of the best estimation.

The optical flow formulation using the two-frame spatial information (TI_DOFE) performs better than optical flow definitions based on only one image information (DOFE_1, DOFE_2) independently to the used multiresolution schema. The new unwarping multiresolution method allows most of the time a better estimation of the displacement field for all kind of optical flow formulations.

However, we can notice that our methods have clearly stronger AAE for motion discontinuity areas. This is due to the MRF formulation of our energy terms that are defined as quadratic functions.

Further results over the all Middlebury optical flow benchmark dataset, including comparisons to other recent techniques are available at the website: http://vision.middlebury.edu/flow/.

5 Conclusion

In this work, we propose a two-frame optical flow formulation using the spatial information from the two images. A new unwarping multiresolution scheme is defined that reduces the number of transformation and interpolation during the pyramidal decreasing process to estimate the displacement field.

Results have shown that the combination of TI_DOFE and noW_MR methods increases the performance of optical flow estimation. The estimation efficiency is as good as state of the art algorithms. It is interesting in a future work to introduce robust function in our MRF framework to be able to extract better motion discontinuities and to define a better data and prior function that physically correspond to the studied motion phenomenon.

References

1. Alvarez, L., Castano, C.A., Garcia, M., Krissian, K., Mazorra, L., Salgado, A., Sanchez, J.: Symmetric Optical Flow. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2007. LNCS, vol. 4739, pp. 676–683. Springer, Heidelberg (2007)
2. Alvarez, L., Weickert, J., Sanchez, J.: Reliable Estimation of Dense Optical Flow Fields with Large Displacements. IJCV 39, 41–56 (2000)
3. Baker, S., Roth, S., Scharstein, D., Black, M.J., Lewis, J.P., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. ICCV 2007, 1–8 (2007), http://vision.middlebury.edu/flow/
4. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of Optical Flow Techniques. IJCV 12, 43–77 (1994)
5. Black, M.J., Anandan, P.: The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow-Fields. CVIU 63, 75–104 (1996)
6. Burt, P.J., Adelson, E.H.: The Laplacian Pyramid as a Compact Image Code. IEEE Trans. on Communications 3, 532–540 (1983)
7. Cassisa, C., Prinet, V., Shao, L., Simoens, S., Liu, C.L.: Optical flow robust estimation in a hybrid multi-resolution MRF Framework. In: ICASSP 2008, Las Vegas (2008)
8. Corpetti, T., Menin, E., Perez, P.: Dense estimation of fluid flow. IEEE Trans. on Pattern Analysis and Machine Intelligence, 365–380 (2002)
9. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. Aritifial Intelligence 17, 185–203 (1981)
10. Liu, H., Rosenfeld, A., Chellapa, R.: Two-frame multi-scale optical flow estimation using wavelet decomposition. In: ICASSP 2002, vol. 3 (2002)
11. Memin, E., Perez, P.: Dense Estimation and Object-Based Segmentation of the Optical-Flow with Robust Techniques. Int. J. on IP 7, 703–719 (1998)
12. Memin, E., Perez, P.: A multigrid approach to hierarchical motion estimation. In: ICCV 1998, January 1998, pp. 933–938 (1998)
13. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly Accurate Optic Flow Computation with Theoretically Justified Warping. IJCV, 141–158 (2006)
14. Papenberg, N., Bruhn, A., Brox, T., Weickert, J.: Numerical Justification for Multiresolution Optical Flow Computation. In: IWCVIA 2003, Las Palmas, Spain (2003)
15. Simoncelli, E.P., Freeman, W.T.: The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation. In: ICIP 1995, pp. 444–447 (1995)
16. Tikhonov, A.N., Arsenin, V.Y.: Solutions of ill-posed Problems, ch. 2. John Wiley & Sons, Chichester (1977)

XIV Video Segmentation and Tracking

Generating Video Textures by PPCA and Gaussian Process Dynamical Model

Wentao Fan and Nizar Bouguila

Institute for Information Systems Engineering
University of Concordia
Montreal, Canada
{wenta_fa,bouguila}@ciise.concordia.ca

Abstract. Video texture is a new type of medium which can provide a continuous, infinitely varying stream of video images from a recorded video clip. It can be synthesized by rearranging the order of frames based on the similarities between all pairs of frames. In this paper, we propose a new method for generating video textures by implementing probabilistic principal components analysis (PPCA) and Gaussian Process Dynamical model (GPDM). Compared to the original video texture technique, video texture synthesized by PPCA and GPDM has the following advantages: it might generate new video frames that have never existed in the input video clip before; the problem of "dead-end" is totally avoided; it could also provide video textures that are more robust to noise.

Keywords: Video texture, computer graphics, computer vision, dimensionality reduction, autoregressive process, Gaussian process, PPCA.

1 Introduction

Video textures, first introduced by Schödl *et al.* [1], is a new type of medium between static image and dynamic video. It can create a continuous, infinitely changing stream of images from a recorded video. Following the work of video texture, Schödl *et al.* also extended this technique on video sprites [2] [3]. Recently, a number of extensions and applications of video texture have emerged. Dong *et al.* [4] proposed a novel method of generating video texture based on wavelet coefficients which are computed from the decomposition of the pixel values of neighboring frames. In the work of Fitzgibbon [5], video texture is synthesized first by applying the principal components analysis (PCA) to obtain the signatures of each frame, then autoregressive process (AR) is used to predict new frames. In [6], Campbell *et al.* extended this approach to work with strongly non-linear sequences.

Our work is inspired from [5], where the author has shown that video texture may be created by implementing regression methods such as AR process which allow the prediction of new video frames. Accordingly, new video textures are obtained by appending synthesized frames. Gaussian process [7] [8] [9] is another approach which can be exploited to solve regression problems. Via Gaussian

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 801–808, 2009.

process, we can define probability distributions over functions directly, and a Gaussian process prior can be combined with a likelihood to acquire a posterior over functions. In our work, we adopt an extension of Gaussian process namely Gaussian process dynamical model (GPDM) [10] [11] which is a latent variable model that can be applied for nonlinear time series analysis. GPDM extended the Gaussian process latent variable model (GPLVM) [12] with a latent dynamical model. In GPDM, it includes a low-dimensional space account for dynamics in the time series data, as well as a mapping from the latent space to observation space. Since video sequence is a time series data, in principle, GPDM is a suitable method to synthesize new video textures. Fitzgibbon [5] has applied PCA as a dimensionality reduction technique to obtain the frames signatures. However, we have shown in a previous works [13] that probabilistic principal components analysis (PPCA) [14] is more robust to noise and provide better results. Thus, our video texture generation framework will be based on both PPCA and GPDMs.

The remainder of this paper is organized as follows. First we introduce Gaussian processes regression in Section 2. Then GPDM for video texture is discussed in Section 3. Section 4 is devoted to the experimental results. The conclusion and future work are included in Section 5.

2 Gaussian Processes Regression

A Gaussian process is defined as a probability distribution over some functions $y(\mathbf{x})$, such that the set of values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\mathbf{x}_1, ..., \mathbf{x}_N$ jointly have a Gaussian distribution. Here, we will illustrate how Gaussian process can be applied on general regression problems. We consider a model where the observed target values t_n are corrupted with some random noise

$$t_n = y_n + \epsilon_n \tag{1}$$

where $y_n = y(\mathbf{x}_n)$ for input data $\mathbf{x}$. ϵ_n is the random noise which has Gaussian distribution with zero mean and β^{-1} variance. Since the noise is independent for each data point, given the values of $\mathbf{y} = (y_1, ...y_N)^T$, the joint distribution of target values $\mathbf{t} = (t_1, ..., t_N)$ is an isotropic Gaussian

$$p(\mathbf{t}|\mathbf{y}) = \mathcal{N}(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}_N) \tag{2}$$

After obtaining the marginal distribution of $\mathbf{t}$, the next job is to evaluate the conditional distribution $p(t_{N+1}|\mathbf{t})$ where t_{N+1} is the next target value that we wish to predict. In order to find $p(t_{N+1}|\mathbf{t})$, we first need to find the joint distribution of $p(\mathbf{t}_{N+1})$ for $t_1, ..., t_{N+1}$

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1}|0, \mathbf{C}_{N+1}) \tag{3}$$

where $\mathbf{C}_{N+1}$ is an $(N+1) \times (N+1)$ covariance matrix. The covariance matrix $\mathbf{C}_{N+1}$ needs to be partitioned as

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \tag{4}$$

where $\mathbf{C}_N$ is the $N \times N$ covariance matrix of the training data, vector $\mathbf{k}$ represents the $N \times 1$ covariance matrix of training data and the predictive target t_{N+1}, and the scalar c denotes the variance of t_{N+1}. As shown in [8], since the joint distribution $p(\mathbf{t}_{N+1})$ is also a Gaussian distribution, we can obtain the mean and covariance of the conditional distribution $p(t_{N+1}|\mathbf{t})$ as

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t} \tag{5}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \tag{6}$$

These results represent the core idea of Gaussian process regression. More details and discussion about Gaussian processes can be found in [7].

3 Gaussian Processes Dynamical Models

The Gaussian process dynamical model (GPDM) [11] is a latent variable model with two nonlinear mappings. One mapping is from the latent space to the observation space and the other is the dynamical mapping in the latent space. Suppose $\{\mathbf{y}_1, ..., \mathbf{y}_N\}$ denotes the D-dimensional observation data set and $\mathbf{y}_t$ represents a particular observation output at the specific time t, $\mathbf{y}_t \in \mathbb{R}^D$. $\mathbf{x}_1, ..., \mathbf{x}_N$ is a data set in the latent space, $\mathbf{x}_t$ represents the d-dimensional latent coordinate of the observation data at time index t, $\mathbf{x}_t \in \mathbb{R}^d$. The first-order Markov dynamics and the latent space mapping are given by

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}; \mathbf{A}) + \mathbf{n}_{x,t} \tag{7}$$

$$\mathbf{y}_t = g(\mathbf{x}_t; \mathbf{B}) + \mathbf{n}_{y,t} \tag{8}$$

here, the dynamical mapping function f is parameterized by $\mathbf{A}$ and latent space mapping function g is is parameterized by $\mathbf{B}$. $\mathbf{n}_{x,t}$ and $\mathbf{n}_{y,t}$ are zero-mean, isotropic, white Gaussian noise processes. Two basis functions ϕ_i and φ_j are used for f and g are given by

$$f(\mathbf{x}; \mathbf{A}) = \sum_i \mathbf{a}_i \phi_i(\mathbf{x}) \tag{9}$$

$$g(\mathbf{x}; \mathbf{B}) = \sum_j \mathbf{b}_j \varphi_i(\mathbf{x}) \tag{10}$$

where weights $\mathbf{A} \equiv [a_1, a_2, ...]^T$ and $\mathbf{B} \equiv [b_1, b_2, ...]^T$. f and g are nonlinear functions of $\mathbf{x}$, but the dependencies of f and g on the parameters $\mathbf{A}$ and $\mathbf{B}$ are linear. For the mapping from latent space to the observation space, after marginalizing over g, the joint distribution of $\mathbf{Y}$ can be represented as

$$p(\mathbf{Y}|\mathbf{X}, \bar{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp(-\frac{1}{2} tr(\mathbf{K}_Y^{-1} \mathbf{Y} \mathbf{W}^2 \mathbf{Y}^T)) \tag{11}$$

here, $\mathbf{K}_Y$ is the kernel matrix of the mapping g and $\bar{\beta}$ are the hyperparameters of the kernel. $\mathbf{W}$ represents the scale parameters which account for the overall scale in each data dimension. The elements of $\mathbf{K}_Y$ are defined by a kernel function $(K_Y)_{ij} \equiv \mathbf{k}_Y(\mathbf{x}_i, \mathbf{x}_j)$. We choose the radial basis function (RBF) as the kernel function for the latent mapping g

$$k_Y(\mathbf{x}, \mathbf{x}') = \beta_1 \exp(-\frac{\beta_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2) + \beta_3^{-1}\delta_{\mathbf{x},\mathbf{x}'} \tag{12}$$

where the hyperparameter β_1 represents the output scale of the kernel function, β_2 represents the inverse width of the RBF, and β_3 gives the variance of the isotropic noise term $\mathbf{n}_{y,t}$. The dynamic mapping for latent coordinate is similar to the latent space mapping. The joint probability density over the latent coordinates can be represent as

$$p(\mathbf{X}|\bar{\alpha}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} \exp(-\frac{1}{2}tr(\mathbf{K}_X^{-1}\mathbf{X}_{2:N}\mathbf{X}_{2:N}^T)) \tag{13}$$

here, $\mathbf{X}_{2:N} = [\mathbf{x}_2, ...\mathbf{x}_N]^T$ denotes the input data that except the first element. $\mathbf{K}_X$ is the kernel matrix build from $[\mathbf{x}_1, ...\mathbf{x}_{N-1}]$. In this dynamic mapping, the form "RBF + linear" is defined for the kernel function

$$k_X(\mathbf{x}, \mathbf{x}') = \alpha_1 \exp(-\frac{\alpha_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2) + \alpha_3 x^T x' + \alpha_4^{-1}\delta_{\mathbf{x},\mathbf{x}'} \tag{14}$$

In order to discourage overfitting, prior distributions are placed on hyperparameters $\bar{\alpha}$, $\bar{\beta}$ and $\mathbf{W}$.[1] Then a generative model for time-series observations can be obtained through a latent space mapping, a dynamic mapping and prior distributions:

$$p(\mathbf{X}, \mathbf{Y}, \bar{\alpha}, \bar{\beta}, \mathbf{W}) = p(\mathbf{Y}|\mathbf{X}, \bar{\beta}, \mathbf{W})p(\mathbf{X}|\bar{\alpha})p(\mathbf{W})p(\bar{\alpha})p(\bar{\beta}) \tag{15}$$

This represents the general form of the GPDM. Details of how to evaluate the parameters for GPDM can be found in [10].

4 Experimental Results

In our work, the goal is to apply GPDM to synthesize video textures. The performance of our approach is evaluated by comparing our results with the video textures generated by AR approach in [5]. In the AR approach for synthesizing video textures, frame signatures are first calculated by adopting the dimension reduction technique: principal components analysis (PCA), followed by the synthesis of new video textures using AR process. In our case, in order to test our approach under different scenarios, several input video clips are selected. First,

[1] $p(\bar{\alpha}) \propto \prod_i \alpha_i^{-1}$, $p(\bar{\beta}) \propto \prod_i \beta_i^{-1}$ and $p(\mathbf{W}) = \prod_{m=1}^{D} \frac{2}{k\sqrt{2\pi}} \exp(-\frac{w_m^2}{2k^2})$, where w_m are the variances that contain the elements of $\mathbf{W}$, and in practice, k is set to 10^3.

the input video clip is decomposed into a sequence of frames. Each individual frame is an input vector $\mathbf{x}$, with dimensionality D. The value of D is the number of pixels contained in each frame. Second, these input vectors are mean-subtracted and the latent coordinates are initialized with PPCA. Last, GPDM is applied to synthesize new video frames which are then composed together to generate a new video texture.

4.1 Generation of New Frames

As described above, new video frames are predicted using GPDM. In other words, it is to predict the next video frame $\mathbf{x}_{N+1}$ conditioned on the previous frame $\mathbf{x}_N$. The marginal distribution of the new frame $p(\mathbf{x}_{N+1})$ derived from the conditional distribution $p(\mathbf{x}_{N+1}|\mathbf{x}_N)$ is also a Gaussian distribution

$$\mathbf{x}_{N+1} \sim \mathcal{N}(\mu_X(\mathbf{x}_N); \sigma_X^2(\mathbf{x}_N)) \tag{16}$$

We can solve this prediction problem by applying the similar ideas as in Gaussian process regression. According to results in (5) and (6), the mean and covariance can be calculated as

$$\mu_X(\mathbf{x}) = \mathbf{X}_{2:N}^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \tag{17}$$

$$\sigma_X^2(\mathbf{x}) = k_X(\mathbf{x}, \mathbf{x}) - \mathbf{k}_X(\mathbf{x})^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \tag{18}$$

In the above equations, $\mathbf{k}_X(\mathbf{x})$ represents a vector that contains the covariance $\mathbf{k}_X(\mathbf{x}, \mathbf{x}_i)$ in the i-th entry and $\mathbf{x}_i$ denotes the i-th training vector. Then, the next frame in the latent space is: $\mathbf{x}_{N+1} = \mu_X(\mathbf{x}_N)$. Therefore, the new video frames can be generated by $\mathbf{y}_{N+1} = \mu_Y(\mathbf{x}_{N+1})$.

New video textures are successfully generated from input video clips by applying PPCA and GPDM with 50 frames in each video texture. They can be played without any visual discontinuity but with similar motions as the original one. Moreover, all resulted frames have never appeared before in the input videos. Fig.1∼ Fig.6 show the first three frames generated by PPCA and GPDM for several input video clips ((a), (b) and (c) represent the first, second and third frame, respectively).

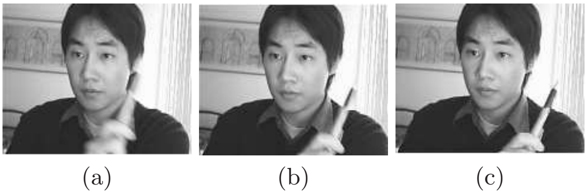


Fig. 1. The first three synthesized frames for a movie of a man moving a pen

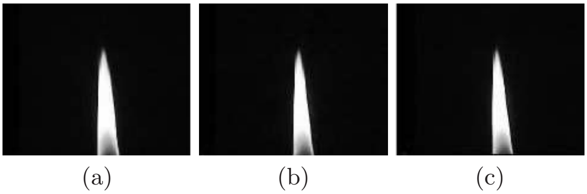


Fig. 2. The first three synthesized frames for a movie a candle flame

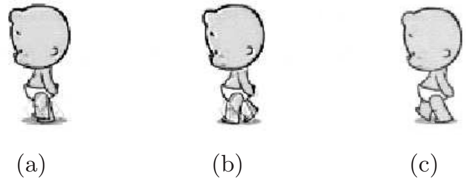


Fig. 3. The first three synthesized frames for an animation of cartoon

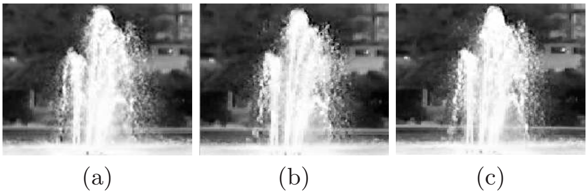


Fig. 4. The first three synthesized frames for a movie of fountain

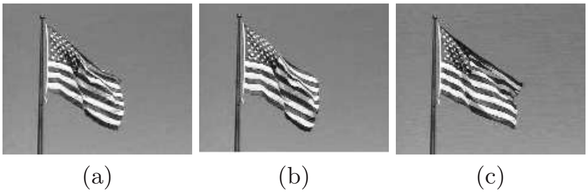


Fig. 5. The first three synthesized frames for a movie of flag

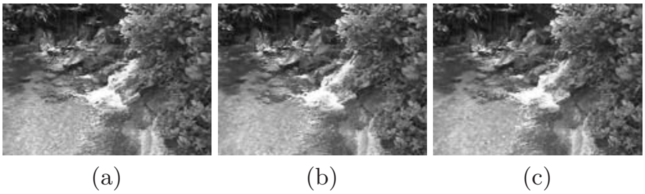


Fig. 6. The first three synthesized frames for a movie of waterfall

4.2 Comparison of the Results

In this section, we compare the performance of synthesizing video textures by GPDM and AR process. Via the AR process, although the result seems very good, there is still one problem which is the occurrence of noise. For all results, after a certain time, the noise will start to become visible and make the video blur. However, through GPDM, it is more robust to noise compared to AR process since it contains a latent space account for the dynamics in the input data. As shown in Fig.7, the 20th, 25th and 30th frames generated by PCA and AR process contain much more noise than the ones produced by PPCA and GPDM at each corresponding frame number. Based on our experimental results, we may conclude that video textures generated by PPCA and GPDM can provide better results with more robustness to noise than AR approach. The synthesized new video textures contain similar motions as the input video clips and all frames in the new video textures are completely new.

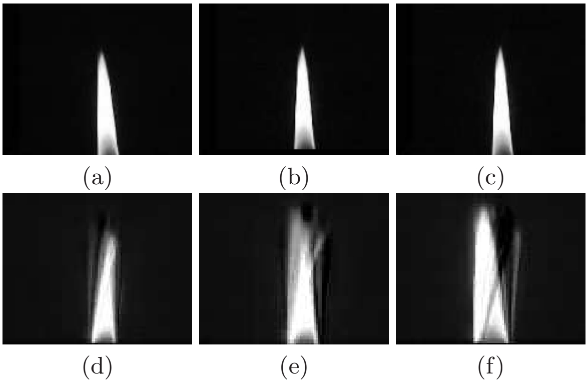


Fig. 7. (a), (b) and (c) illustrate the 20th, 25th and 30th frames synthesized by PPCA and GPDM; (d), (e) and (f) demonstrate the 20th, 25th and 30th frames generated by PCA and AR process

5 Conclusion and Future Works

In this paper, we proposed a new approach for generating video textures using PPCA and GPDM. GPDM is a nonparametric model for learning high-dimensional nonlinear dynamical data sets. We have tested PPCA and GPDM on several movie clips, it can generate video textures containing frames that never appeared before with similar motions as the original video. Compared with PCA and AR process, PPCA and GPDM can produce better results with more robustness to noise. Unfortunately, video textures synthesized by PPCA and GPDM still have visual discontinuities for some highly structured and variable motions (such as dancing and fighting). Thus, there might be some more potential improvements on generating video textures. Since GPDM is highly dependent on the kernel functions, selection of a better function would be a key factor for improving the predictive power. Besides this, We also would like to modify the statistical model of the GPDM in order to acquire the ability of modelling highly variable motion sequences in the future.

Acknowledgments. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC) and a NATEQ Nouveaux Chercheurs Grant. Finally, the authors thank CIARP reviewers for their constructive comments.

References

1. Schödl, A., Szeliski, R., Salesin, D., Essa, I.: Video textures. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 489–498. ACM Press/Addison-Wesley Publishing Co., New York (2000)
2. Schödl, A., Essa, I.: Machine learning for video-based rendering. Advances in Neural Information Processing Systems 13, 1002–1008 (2001)
3. Schödl, A., Essa, I.: Controlled animation of video sprites. In: ACM SIGGRAPH Symposium on Computer Animation, pp. 121–128 (2002)
4. Dong, J.Y., Zhang, C.X., Wang, Y.J., Zhou, H.Y.: A video texture synthesis method based on wavelet transform. In: Wavelet Analysis and Pattern Recognition, ICWAPR 2007. International Conference, Beijing, pp. 1177–1181 (2007)
5. Fitzgibbon, A.W.: Stochastic rigidity: image registration for nowhere-static scenes. In: Proceedings of International Conference on Computer Vision (ICCV) 2001, vol. 1, pp. 662–669 (2001)
6. Campbell, N., Dalton, C., Gibson, D., Thomas, B.: Practical generation of video textures using the auto-regressive process. Image and Vision Computing 22, 819–827 (2004)
7. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2006)
8. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)
9. MacKay, D.J.C.: Introduction to Gaussian processes. Neural Networks and Machine Learning 168, 133–165 (1998)
10. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 283–298 (2008)
11. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: Advances in Neural Information Processing Systems 18, pp. 1441–1448. MIT Press, Cambridge (2006)
12. Lawrence, N.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. The Journal of Machine Learning Research 6, 1783–1816 (2005)
13. Fan, W., Bouguila, N.: On Video Textures Generation: A comparison Between Different Dimensionality Reduction Techniques. In: IEEE International Conference on System, Man, and Cybernetics (2009)
14. Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B 61, 611–622 (1999)

Fuzzy Feature-Based Upper Body Tracking with IP PTZ Camera Control

Parisa Darvish Zadeh Varcheie and Guillaume-Alexandre Bilodeau

Department of Computer Engineering and Software Engineering,
École Polytechnique de Montréal,
P.O. Box 6079, Station Centre-ville Montréal (Québec), Canada, H3C 3A7
{parisa.darvish-zadeh-varcheie,
guillaume-alexandre.bilodeau}@polymtl.ca

Abstract. In this paper, we propose a fuzzy-feature based method for online upper body tracking using an IP PTZ camera. Because the camera uses a built-in web server, camera control entails camera response time and network delays, and thus, the frame rate is irregular and in general low (2-7 fps). It detects at every frame, candidate targets by extracting motion, a sampling method, and appearance. The target is detected among samples with a fuzzy classifier. Results show that our system has a good target detection precision ($> 85\%$), low track fragmentation, and the target is almost always localized within 1/6th of the image diagonal from the image center.

1 Introduction

People detection and tracking are important capabilities for applications that desire to achieve a natural human-machine interaction such as people identification. Here, we are interested in human upper body tracking by an IP PTZ camera (a network-based camera that pans, tilts and zooms). Upper body tracking determines the location of the upper body in each image. An IP PTZ camera communicates and responds to command via its integrated web server after some delays. Tracking with such camera involves some difficulties which are: 1) irregular response time to control command, 2) low usable frame rate (while the camera executes the motion command, the frames received are useless), 3) irregular frame rate because of network delays (the time between two frames is not necessarily constant), 4) changing field of view (FOV) resulting from panning, tilting and zooming and 5) various scales of objects.

Much works on face and upper body tracking have been reported. Comaniciu *et al.* [1] applied the mean-shift algorithm to an elliptical region which is modeled by histogram for face tracking. They also take advantage of the gradient perpendicular to the border of the hypothesized face region and background subtraction. This method is not designed to cope with large motion. The algorithm in Ido *et al.* [2] works by maximizing the PDF of the target's bitmap, which is formulated by the color and location of pixel at each frame. Severe occlusions are not handled and this algorithm is not very fast. Roha *et al.* [3] proposed a contour-based object tracking method using optical flow. It has been tested by selecting tracked object boundary edges in a video stream with a changing background and a moving camera. The face region needs to be large and it

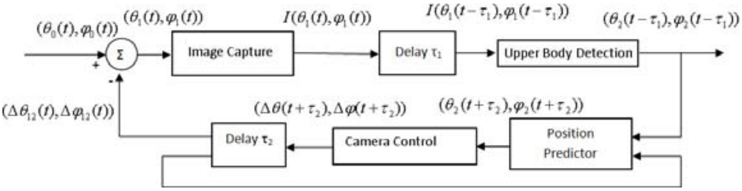
E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 809–816, 2009.

is computationally expensive. In the work of Elder *et al.* [4] a stationary, preattentive, low-resolution wide FOV camera, and a mobile, attentive, high-resolution narrow FOV camera are used. They used skin detection, motion detection and foreground extraction for face tracking. The advantage of this work is a wide FOV, but it relies on a communication feedback between two cameras. Funahasahi *et al.* [5] developed a hierarchical tracking method using a stationary camera and a PTZ camera. The face needs to be large enough to detect the irises. Then, detected irises are used as feature for face detection. In the method of Bernardin *et al.* [6] the upper body histogram information, KLT feature tracker, and active camera calibration are combined to track the person for 3D localization application. In the algorithm of Li *et al.* [7] each observer should be learned from different ranges of samples, with various subsets of features. Learning step is based on model complexity and increases computation time. The method has limitations in distinguishing between different targets, and has model overupdating problems. Kang *et al.* [8] uses a geometric transform-based mosaicing method for person tracking by a PTZ camera. For each consecutive frame, it finds the good features for the correspondence and then tries to shift the moved image. They are using a high cost background modeling using a calibration scheme, which is not suitable for tracking by internet-based PTZ cameras.

In our work, we want to cope with the problem of large motion detection, low usable frame rate, and tracking with various scale changes. In addition, the tracking algorithm should handle the camera response time. The proposed method consists of target modeling to represent the tracked object, target candidates detection (sampling), target localization using a fuzzy classifier, target position prediction and camera control to center the PTZ camera on the target. Results show that our system has a good target detection precision ($> 85\%$), low track fragmentation, and the target is almost always localized within 1/6th of the image diagonal from the image center.

2 System Architecture and Methodology

The servo controlling and tracking system is modeled by a closed-loop control which has a negative feedback as shown in Fig. 1. It consists of three main blocks : image capture, upper body detection and camera control. Tracking is affected by two delays which are the delay from image capture and the delay in the feedback loop from executing camera motion commands. The delay from upper body detection is considered negligible compared to the two other delays. The input of the system is the current pan



$(\theta_0(t),\varphi_0(t))$
$(\theta_1(t),\varphi_1(t))$
$I(\theta_1(t),\varphi_1(t))$
$I(\theta_1(t-\tau_1),\varphi_1(t-\tau_1))$
$(\theta_2(t-\tau_1),\varphi_2(t-\tau_1))$
Σ
+
-
Image Capture
Delay τ1
Upper Body Detection
$(\Delta\theta_{12}(t),\Delta\varphi_{12}(t))$
$(\Delta\theta(t+\tau_2),\Delta\varphi(t+\tau_2))$
$(\theta_2(t+\tau_2),\varphi_2(t+\tau_2))$
Delay τ2
Camera Control
Position Predictor

Fig. 1. The system architecture and servo control model. (θ_0,ϕ_0):initial pan-tilt angles, $(\Delta\theta_{12},\Delta\phi_{12})$ means $(\theta_1$-θ_2,ϕ_1-$\phi_2)$.

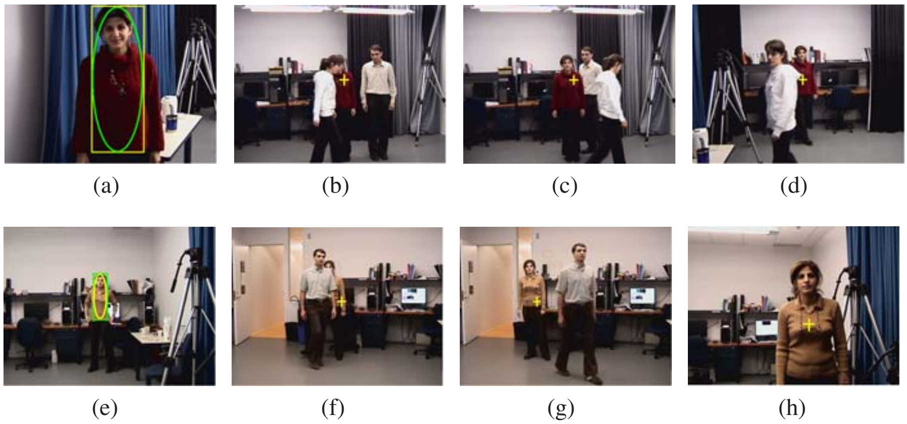
and tilt angles of the camera and the output will be the determined pan and tilt angles by the fuzzy classifier. The delays imply the current position of the target cannot be used for centering the camera. The algorithm in upper body detection has three steps: 1) target modeling to represent the tracked object, 2) target candidates detection (sampling), and 3) target localization by scoring sample features. Target position prediction and camera control utilize the upper body detection results. To compensate for motion of the target during the delays, a position predictor block is added. Camera control is used to put the image center of PTZ camera on the target. We have made the following assumptions: 1) skin detection will be done over the yellow, white, light brown and pink color skin types from number 1 to 26 on Von Luschans skin chromatic scale (almost 73% of all skin types in the world [9]), 2) persons walk at a normal pace or fast, but do not run, 3) the target person can walk in any direction, but the face should be always visible partially, 4) a wide FOV (approximately $48\,^{\circ}$) is assumed and scenes are not crowded (max 2-3 persons).

2.1 Upper Body Detection

Target modeling: A target is represented by an elliptical image region. It is modeled by two features: 1) quantized HSV color histogram with 162 bins (i.e. $18 \times 3 \times 3$) and 2) the mean of R, G and B color components of RGB color space of all the pixels inside of the elliptical region. Initialization is done manually by selecting the top part of the body (head and torso) of the person. We fit an ellipse inside the bounding box of the selected region (Fig. 2 (a) and (e)). Ellipse fits better the shape of the head and torso. Then the initial target M is modeled by the two discussed features.

Target candidates detection (sampling): For tracking, we sample with ellipse the image around regions of interest, model them and filter them. There are two regions of interest: 1) areas with motion, 2) the center of the image.

1. *Motion-based samples*: The first type of samples is detected using motion of the target from the difference of two consecutive frames while the camera is not moving. The difference results are noisy and some morphological operations such as erosion, dilation, image closing (by a circular structuring element of 3 pixels radius) and filtering (by a 3×3 median filter) are used to reduce noise. Whenever a moving object in the scene has a color similar to the background or has an overlap with its previous frame position, some parts of the moving object are not detected as foreground regions. This results in detecting smaller regions that are fragments of a larger one. Fragments are merged iteratively based on their proximity. The small regions that are nearby, and whose contours are in intersection, are merged. A motion-based sample is an ellipse which circumscribes an area with motion.
2. *Fixed samples*: According to our goal, the object should be always near the image center. To have robust tracking even when there is no motion from the target, we consider F additional fixed samples in the image which are generated by a uniform function and located around the center (typically $F = 16$). Samples are in large and small sizes. The largest sample is used for zooming or for object approaching the camera. Its area is 1/3 of the image area. The small samples are used for a target far from the camera and close to the center in different positions. The sizes of these


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. Examples of tracking frames for Exp_1 (a) to (d) and Exp_8 (e) to (h). Exp_1 (a) initial model selection, (b) short-term occlusion, (c) after occlusion, (d) scale variation; Exp_8 (e) initial model selection, (f) short-term occlusion, (g) after occlusion, (h) scale variation.

elliptic samples are obtained experimentally according to the minimum face size, which is in our algorithm 5x5 pixels from frontal and lateral views.

3. *Blob filtering and modeling*: Our targets of interest are persons, thus all samples are filtered by a Bayesian skin classifier which has high true positive rate, simple implementation and minimum cost [10]. All the skin regions that contain less than 40 skin pixels (less than the half of the minimum face size) or do not contain skin regions are removed. The torso is assumed to be below the detected skin region and two times longer than the height of detected skin region. Thus, the ellipse width is the same as the skin region width, and its height is three times longer than the skin region height.

Sample likelihood using a fuzzy classifier: To localize the target, features of each sample S_i are compared with the initial model M, and a score ($ScoreS_i$) as a sample liklihood is given to each S_i using a fuzzy rule. The score is obtained by multiplying four fuzzy membership functions which will be explained in the following.

$$ScoreS_i = \mu_{EC}.\mu_{EP}.\mu_{EH}.\mu_H. \qquad (1)$$

The target is the sample with the highest score. We are using four membership functions, each with fuzzy outputs between 0 and 1:

1. The membership function μ_{EC} is used for Euclidean distance of mean RGB of S_i (R_{si}, G_{si}, B_{si}) with the mean RGB of M (R_m,G_m, B_m). It is defined as

$$\mu_{EC} = 1 - \frac{\sqrt{(R_{si} - R_m)^2 + (G_{si} - G_m)^2 + (B_{si} - B_m)^2}}{255\sqrt{3}}. \qquad (2)$$

2. The membership function μ_{EP} is utilized for Euclidean distance of S_i centroid, (x_{si}, y_{si}), from the image center,(x_{im}, y_{im}). Indeed normally, the person should be near the image center. σ^2 is equal to a quarter of the image area around the image center. It is defined as

$$\mu_{EP} = exp(-\frac{(\sqrt{(x_{si} - x_{im})^2 + (y_{si} - y_{im})^2})^2}{2\sigma^2}). \tag{3}$$

3. The membership function (μ_{EH}) is applied for Euclidean distance of normalized quantized HSV color histogram of S_i, H_{si}, with the histogram of M, H_m, with n histogram bins [11]. It is computed as

$$\mu_{EH} = 1 - \sqrt{\frac{\sum_n (H_{si}[n] - H_m[n])^2}{2}}. \tag{4}$$

4. Finally the membership function of μ_H is used for similarity of normalized quantized HSV color histogram of S_i with histogram of M with average of normalized histograms of $\bar{H}_{si}$ and $\bar{H}_m$ respectively [12]. It is the normalized correlation coefficient of two histograms and is defined as

$$\mu_H = \frac{1}{2} + \frac{\sum_n ((H_{si}[n] - \bar{H}_{si})(H_m[n] - \bar{H}_m))}{2 \times \sqrt{\sum_n (H_{si}[n] - \bar{H}_{si})^2} \sqrt{\sum_n (H_m[n] - \bar{H}_m)^2}}. \tag{5}$$

2.2 Target Position Prediction and Camera Control

As discussed in Section 2, a position predictor based on the two last motion vectors has been designed to compensate for motion of the target during the delays. This motion predictor will consider the angle between two consecutive motion vectors. If the angle difference is smaller than $25\,^\circ$, it is assumed the target is moving in the same direction. Thus the system will put the camera center on the predicted position which is :

$$x_P = x_E + \bar{\tau}_2 \times \frac{\Delta x_1 + \Delta x_2}{\tau_1^1 + \tau_1^2}. \tag{6}$$

where Δx_1 and Δx_2 are the two target displacement vectors (i.e. target motion vector). τ_1^1, τ_1^2 are delay τ_1 between two last captured images. x_P is the predicted target coordinate and x_E is the extracted target coordinate from the fuzzy classifier. $\bar{\tau}_2$ is the average delay time τ_2 obtained from previous camera movements. To follow the target, the PTZ motors are commanded based on x_P. Camera is controlled by computing the pan and tilt angles from a workstation and sending HTTP POST request using the CGI scripts of the camera [13].

3 Results and Discussion

We used one Sony IP PTZ camera (SNC-RZ50N) for our tests. For validation, we tested the complete system in online experiments. The algorithm is implemented on an Intel

Xeon(R) 5150 in C++ using OpenCV. The tracking algorithm has been tested over events such as entering or leaving the FOV of the camera and occlusion with other people in the scene. We recorded all the experiments to extract their ground-truth manually for performance evaluation. In the general scenario of the experiments, a target actor from the frontal view is selected for initial modeling. She starts to walk around in a room. Two or three actors can walk at the same time in different directions, crossing and occluding with the target. Fig. 2 shows the initial model selection and some frames obtained during tracking. We have done ten experiments with the IP camera and experiments are classified into two classes based on the image resolution as described in table 2. To evaluate our method, four metrics are used as explained in table 1.

Table 1. Evaluation metrics

Metric	Description
$P = \frac{TP}{TP+FP}$	to calculate the target localization accuracy
$d_{gc} = \frac{\sqrt{(x_c-x_g)^2+(y_c-y_g)^2}}{a}$	to evaluate the dynamic performance of the tracking system; It is the spatial latency of the tracking system, as ideally, the target should be at the image center.
$d_{gp} = \frac{T_{OUT}}{NF}$	to evaluate the error of tracking algorithm. Ideally, d_{gp} should be zero.
$TF = \frac{TP}{TP+FP}$	to indicate the lack of continuity of the tracking system for a single target track [14]

TP: number of frames with target located correctly, FP:number of frames with target not located correctly, a: radius of circle which circumscribes the image, (x_g,y_g): ground-truth target coordinate, (x_c,y_c): image center, (x_p,y_p): tracked object coordinate, T_{OUT}: number of frames with target out of FOV, NF: total number of frames.

Table 2 shows the results of the four metrics with the system frame rate for all experiments. For d_{gc} and d_{gp}, we show the mean and variance of all experiments. For class 1, because of the lower system frame rate, the method has lost the target several times, but eventually recovers. Because of d_{EP} and camera control, the error on $\mu_{d_{gc}}$ has effect on $\mu_{d_{gp}}$ and vice versa. For class 2, the system frame rate is increased because of smaller image size. Thus TF for class 2 is smaller than class 1. A faster system frame rate improves the results of TF, $\mu_{d_{gc}}$, $\mu_{d_{gp}}$ and P. The last two columns in Table 2 are the minimum and maximum length of target motion vector in number of pixels. Results show that our algorithm can handle and overcome large motion (i.e. high values of $max(\varDelta x)$) because of using a repetitive target detection scheme and motion prediction technique that does not rely on spatial proximity. It will lose a target only if the target changes its motion direction suddenly and walks very fast in the opposite predicted position (e.g. experiments with $TF \neq 0$). By using a repetitive detection scheme and combining it with a motion predictor, we can handle random motion between frames, as long as the target position is well predicted, and its appearance does not change significantly. The motion predictor is used to compensate the two delays τ_1 and τ_2 discussed in Section 2, which may cause the target to exit the FOV.

Table 2. Experimental results

	P (%)	TF (%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	$FR(fps)$	NF	Δx_{min}	Δx_{max}	IS	IMP
E_1	97	2.3	0.1834	0.0296	0.0573	0.0047	3.02	563	10	246	L	N
E_2	96	1.5	0.1180	0.0170	0.0232	0.0009	2.98	472	1	194	L	N
E_3	88	1.8	0.2462	0.0341	0.0774	0.0043	2.75	524	3	306	L	F
E_4	92	2.1	0.1427	0.0209	0.0420	0.0018	2.64	602	2	562	L	F
E_5	70	1.7	0.3194	0.0618	0.0690	0.0044	3.1	578	12	391	L	M
class 1	**88**	**1.9**	**0.2019**	**0.0327**	**0.0538**	**0.0032**	**2.88**	**2739**	**1**	**562**	L	A
E_6	94	0.7	0.1597	0.0354	0.0302	0.0019	6.18	908	1	242	S	N
E_7	100	0	0.1101	0.0156	0.0395	0.0022	6.22	964	0	184	S	N
E_8	100	0	0.0997	0.0127	0.0215	0.0012	6.87	889	2	152	S	F
E_9	93	0.4	0.1210	0.0368	0.0282	0.0026	6.69	952	0	154	S	F
E_{10}	90	0.2	0.3577	0.0341	0.0877	0.0032	6.97	994	2	255	S	M
class 2	**95**	**0.26**	**0.1696**	**0.0269**	**0.0414**	**0.0022**	**6.57**	**4707**	**0**	**255**	S	A

$\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and Δx_{min} and Δx_{max}: minimum and maximum motion vector length, IS: Image Size, IMP: Initial model position from camera, N: Near, F: Far, M: Middle, L: 640 x 480, S: 320 x 240, A: All possible initial model positions from camera.

Generally, according to the mean of distances, the location of the target is near to the ground-truth. The target is usually localized within 1/6th of the image diagonal from the image center. With faster system frame rate the results of tracking have been improved significantly. When localization fails, it is because of similarity or closeness of the color histogram of the target with other blobs. The image resolution has effect on the system frame rate and thus on tracking error. In all experiments, there are scale changes to verify tracking against scaling. Our algorithm can overcome scaling variations even in the image with minimum 5×5 face size (e.g. Fig.2(e) and (d)). It is because of using normalized color histogram and average color features. These two features are independent of the size of the target. Our method can also recover the tracking if it loses the object (e.g. experiments with $TF \neq 0$), because of the repetitive detection scheme. Of course, it is conditional to the object being in the FOV of the camera. Occlusions are handled in the same way. However, when the object is occluded, another similar object will be tracked (the most likely candidate blob) until the occlusion ends. This could cause the real target to become out of the FOV of the camera. Fig. 2 shows an example of short-term occlusion handling. The proposed method can handle it in this case. In the reported experiments, occlusion did not cause difficulties. The duration of the experiments is short because the goal the algorithm will be zooming on target face and capturing it for identification purpose.

4 Conclusion

In this paper, an upper body tracking algorithm for IP PTZ camera in online application is proposed. The proposed method consists of three main parts: image capture, upper body detection and camera control. We use a fuzzy classifier because our system has uncertainty and is nonlinear. Results show that our algorithm can handle and overcome large motion between two consecutive frames, because it is based on combination of re-detecting the target and target position prediction at each frame. We will lose a target if the person changes its motion direction suddenly and walks very fast in the opposite predicted direction. We can recover the track if the target moves inside the FOV of the camera again. The proposed method can handle indirectly the short-term occlusion at the condition that the object stays in the FOV. We get better results with faster system frame rate.

Future work of the method will be adding camera zooming and enhancing robustness of the motion prediction to prevent the target being out of the camera FOV.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE T-PAMI 25(5), 564–577 (2003)
2. Leichter, I., Lindenbaum, M., Rivlin, E.: Bittracker- a bitmap tracker for visual tracking under very general conditions. IEEE T-PAMI 30(9), 1572–1588 (2008)
3. Roha, M., Kima, T., Park, J., Lee, S.: Accurate object contour tracking based on boundary edge selection. Pattern Recognition 40(3), 931–943 (2007)
4. Elder, J.H., Prince, S., Hou, Y., Sizintsev, M., Olevsky, E.: Pre-attentive and attentive detection of humans in wide-field scenes. International Journal of Computer Vision 72(1), 47–66 (2007)
5. Funahashi, T., Tominaga, M., Fujiwara, T., Koshimizu, H.: Hierarchical face tracking by using ptz camera. In: IEEE Int. Conf. on Automatic Face and Gesture Recognition (FGR), pp. 427–432 (2004)
6. Bernardin, K., Camp, F., Stiefelhagen, R.: Automatic person detection and tracking using fuzzy controlled active cameras. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
7. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans. IEEE T-PAMI 30(10), 1728–1740 (2008)
8. Kang, S., Paik, J., Koschan, A., Abidi, B., Abidi, M.: Real-time video tracking using ptz cameras. In: 6th Int. Conf. on Quality Control by Artificial Vision, pp. 103–111 (2003)
9. Wikipedia: Von luschan's chromatic scale — wikipedia, the free encyclopedia (2008), http://en.wikipedia.org/w/index.phptitle=Von_Luschan%27s_chromatic_sca%le&oldid=249213206 (Online; accessed June 9, 2009)
10. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognition 40(3), 1106–1122 (2007)
11. Cha, S.H., Srihari, S.N.: On measuring the distance between histograms. Pattern Recognition 35(6), 1355–1370 (2002)
12. Boufama, B., Ali, M.: Tracking multiple people in the context of video surveillance. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 581–592. Springer, Heidelberg (2007)
13. Sony corporation: Snc-rz25n/p cgi command manual, version 1.0 (2005)
14. Yin, F., Makris, D., Velastin, S.: Performance evaluation of object tracking algorithms. In: IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance(PETS) (2007)

Experimental Assessment of Probabilistic Integrated Object Recognition and Tracking Methods

Francesc Serratosa[1], Nicolás Amézquita[1], and René Alquézar[2]

[1] Universitat Rovira i Virgili
Av. Països Catalans 26, 43007 Tarragona, Spain
[2] Inst. Robòtica i Informàtica Industrial
CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain
francesc.serratosa@urv.cat, nicolas.amezquita@urv.cat,
ralquezar@iri.upc.edu

Abstract. This paper presents a comparison of two classifiers that are used as a first step within a probabilistic object recognition and tracking framework called PIORT. This first step is a static recognition module that provides class probabilities for each pixel of the image from a set of local features. One of the implemented classifiers is a Bayesian method based on maximum likelihood and the other one is based on a neural network. The experimental results show that, on one hand, both classifiers (although they are very different approaches) yield a similar performance when they are integrated within the tracking framework. And on the other hand, our object recognition and tracking framework obtains good results when compared to other published tracking methods in video sequences taken with a moving camera and including total and partial occlusions of the tracked object.

Keywords: Object tracking, object recognition, occlusion, performance evaluation.

1 Introduction

The first important issue while dealing with object locating and tracking is to determine the type of object model to learn, which usually depends on the application environment. In our case, we want a mobile robot equipped with a camera to locate and track general objects (people, other robots, wastepaper bins…) in both indoor and outdoor environments.

On one hand, a useful model should be relatively simple and easy to acquire from the result of image processing steps. For instance, the result of a color image segmentation process, consisting of a set of regions or spots, characterized by simple features related to color, may be a good starting point to learn the model. Hence, we have decided to represent an object just as an unstructured set of pixels.

On the other hand, we want the system to have the capacity of determining occlusions and re-emergencies of tracked objects. Various approaches that analyze occlusion situations have been proposed. The most common one is based on background subtraction [1]. Although this method is reliable, yet it only works with a fixed camera and a known background, which is not our case. Other approaches are based on

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 817–824, 2009.

examining the measurement error for each pixel [2, 3]. The pixels that their measurement error exceeds a certain value are considered to be occluded. These methods are not very appropriate in outdoor scenarios, where the variability of the pixel values between adjacent frames may be high. Finally, contextual information is exploited in [4, 5], but in these approaches, there is a need of knowing a priori the surroundings of the mobile robot.

This paper presents a comparison of two possible alternative classifiers to deal with the first step of a previously reported approach for integrated object recognition and tracking [6, 7]. These are a simple Bayesian method and a neural net based method, both providing posterior class probabilities for each pixel of the images.

The rest of the paper is organized as follows. A summary of our probabilistic framework for object recognition and tracking is given in Section 2. The methods used for the static recognition module are described in Section 3. Experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 A Probabilistic Framework for Object Recognition and Tracking

Let us assume that we have a sequence of 2D color images $I^t(x,y)$ for $t=1,\dots,L$, and that there are a maximum of N objects of interest in the sequence of different types (associated with classes $c=1,\dots,N$), and that a special class $c=N+1$ is reserved for the background. Hence, we would like to obtain N sequences of binary images $T_c^t(x,y)$, that mark the pixels belonging to each object in each image; these images are the desired output of the whole process and can also be regarded as the output of a tracking process for each object.

In our PIORT (Probabilistic Integrated Object Recognition and Tracking) framework [6, 7], we divide the system in three modules. The first one performs object recognition in the current frame (**static recognition**) and stores the results in the form of probability images (one probability image per class) $Q_c^t(x,y)$, that represent for each pixel the probabilities of belonging to each one of the objects of interest or to the background, according only to the information in the current frame (see Section 3). In the second module (**dynamic recognition**), the results of the first module are used to update a second set of probability images, $p_c^t(x,y)$, with a meaning similar to that of $Q_c^t(x,y)$ but now taking into account as well both the recognition and tracking results in the previous frames through a dynamic iterative rule. Finally, in the third module (**tracking decision**), tracking binary images $T_c^t(x,y)$ are determined for each object from the current dynamic recognition probabilities, the previous tracking image of the same object and some other data, which contribute to provide a prediction of the object's apparent motion in terms of translation and scale changes. See [6] for a detailed description of the second and third modules and [7] for an extension of the tracking decision module.

3 Static Recognition Module

The static recognition module in our PIORT framework is based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The local features to be used may be chosen in many different ways. A possible approach consists of first segmenting the given input image $I^{t}(x,y)$ in homogeneous regions (or spots) and computing some features for each region that are afterwards shared by all its constituent pixels. Hence, the class probabilities $Q_c^t(x,y)$ are actually computed by the classifier once for each spot in the segmented image and then replicated for all the pixels in the spot. For instance, RGB colour averages can be extracted for each spot after colour segmentation and used as feature vector *v(x,y)* for a classifier. In the next two subsections we present two specific classifiers that have been implemented and tested within the PIORT framework using this type of information.

3.1 A Simple Bayesian Method Based on Maximum Likelihood and Background Uniform Conditional Probability

Let *c* be an identifier of a class (between 1 and *N*+1), let *B* denote the special class *c*=*N*+1 reserved for the background, let *k* be an identifier of an object (non-background) class between 1 and *N*, and let *v* represent the value of a feature vector. Bayes theorem establishes that the posterior class probabilities can be computed as

$$P(c \mid v) = \frac{P(v \mid c)P(c)}{P(v)} = \frac{P(v \mid c)P(c)}{P(v \mid B)P(B) + \sum_{k=1}^{N} P(v \mid k)P(k)} \tag{1}$$

Our simple Bayesian method for static recognition is based on imposing the two following assumptions:

a) equal priors: all classes, including *B*, will have the same prior probability, i.e. *P*(*B*)=1/(*N*+1) and *P*(*K*)=1/(*N*+1) for all *k* between 1 and *N*.
b) a uniform conditional probability for the background class, i.e. *P*(*v*|*B*)=1/*M*, where *M* is the number of values (bins) in which the feature vector *v* is discretized.

Note that the former assumption is that of a maximum likelihood classifier, whereas the latter assumes no knowledge about the background. After imposing these conditions, equation (1) turns into

$$P(c \mid v) = \frac{P(v \mid c)}{\frac{1}{M} + \sum_{k=1}^{N} P(v \mid k)} \tag{2}$$

and this gives the posterior class probabilities we assign to the static probability images, i.e. $Q_c^t(x,y) = P(c \mid v(x,y))$ for each pixel (*x*,*y*) and time *t*.

It only remains to set a suitable *M* constant and to estimate the class conditional probabilities *P*(*v* | *k*) for all *k* between 1 and *N* (object classes). To this end, class histograms H_k are set up using the labelled training data and updated on-line afterwards using the tracking results in the test data.

For constructing the histograms, let *v(x,y)* be the feature vector consisting of the original RGB values of a pixel (*x*,*y*) labelled as belonging to class *k*. We uniformly discretize each of the R, G and B channels in 16 levels, so that M =16×16×16= 4096. Let b be the bin in which $v(x,y)$ is mapped by this discretization. To reduce discretization effects, a smoothing technique is applied when accumulating counts in the histogram as follows:

$$\begin{aligned} &H_k(b) := H_k(b) + (10 - \#neighbors(b)) \\ &H_k(b') := H_k(b') + 1 \quad \text{if } b' \text{ is a neighbor of } b \end{aligned} \tag{3}$$

where the number of neighbors of b (using non-diagonal connectivity) varies from 3 to 6, depending on the position of b in the RGB space. Hence, the total count C_k of the histogram is increased by ten (instead of one) each time a pixel is counted and the conditional probability is estimated as $P(v \mid k) = H_k(b) / C_k$ where b is the bin corresponding to v. The above smoothing technique is also applied when updating the histogram from the tracking results; in that case the RGB value $v(x,y)$ in the input image $I^t(x,y)$ of a pixel (x,y) is used to update the histogram H_k (and the associated count C_k) if and only if $T_k^t(x,y)$=1.

3.2 A Neural Net Based Method

In this method, a neural net classifier (a multilayer perceptron) is trained off-line from the labelled training data. The RGB colour averages extracted for each spot after colour segmentation are used as feature vector $v(x,y)$ and supplied as input to the network in both training and test phases. To the contrary of the Bayesian method described previously, training data for the background class are also provided by selecting some representative background regions in the training image sequence, because the network needs to gather examples for all classes including the background. The network is not retrained on-line using the tracking results in the test phase (this is another difference with respect to the Bayesian method described).

It's well known that using a 1-of-c target coding scheme for the classes, the outputs of a network trained by minimizing a sum-of-squares error function approximate the posterior probabilities of class membership (here, $Q_c^t(x,y)$), conditioned on the input feature vector [8]. Anyway, to guarantee a proper sum to unity of the posterior probabilities, the network outputs (which are always positive values between 0 and 1) are divided by their sum before assigning the posterior probabilities.

4 Experimental Results

We were interested in testing both PIORT approaches in video sequences taken with a moving camera and including object occlusions. To this end, we have used three test sequences with N=1 objects of interest to track, which are available at http://www-iri.upc.es/people/ralqueza/S5.avi, S8.avi and S9.avi, respectively. The first sequence shows an office scene where a blue ball is moving on a table and is temporally occluded, while other blue objects appear in the scene. A similar but different sequence was used for training a neural network to discriminate between blue balls and typical sample regions in the background and for constructing the class histogram of the blue ball (available at http://www-iri.upc.es/people/ralqueza/bluetraining.avi). The second sequence is a long sequence taken on a street where the aim is to track a pedestrian wearing a red jacket and which includes total and partial occlusions of the followed person. In this case, a short sequence of the scene taken with a moving camera located in a different position was used as training sequence (http://www-iri.upc.es/people/ralqueza/redpedestrian_training.avi). The third sequence, S9.avi, is even longer and shows an outdoor scene in which a guy riding a Segway robot and wearing an orange T-shirt is followed; the associated training sequence is at http://www-iri.upc.es/people/ralqueza/T-shirt_training.avi.

All images in the sequences were segmented independently using the EDISON implementation of the mean-shift segmentation algorithm, code available at http://www.caip.rutgers.edu/riul/research/code.html. The local features extracted for each spot of each image were the RGB colour averages of the pixels in that spot. For object learning, spots selected through ROI (region-of-interest) windows in the training sequence were collected to train a two-layer perceptron using backpropagation and to build the target class histogram. When using the neural net in the test phase, the class probabilities for all the spots in the test sequences were estimated from the net outputs. When using the histogram, the spot class probabilities were estimated according to equation (2). In both cases, the spot class probabilities were replicated for all the pixels in the same spot. For object tracking in the test sequences, ROI windows for the target object were only marked in the first image to initialise the tracking process.

The results for the test sequences were stored in videos where each frame has a layout of 2 x 3 images with the following contents: the top left is the image segmented by EDISON; the top middle is the image of probabilities given by the static recognition module for the current frame; the top right is the *a priori* prediction of the tracking image; the bottom left is the image of dynamic probabilities; the bottom right is the *a posteriori* binary tracking image (the final result for the frame); and the bottom middle is an intermediate image labelled by the tracking module where yellow pixels correspond to pixels labelled as "certainly belonging to the object", light blue pixels correspond to pixels initially labelled as "uncertain" but with a high dynamic probability, dark blue pixels correspond to pixels labelled as "uncertain" and with a low probability, dark grey pixels are pixels labelled as "certainly not belonging to the object" but with a high probability and the rest are black pixels with both a low probability and a "certainly not belonging to the object" label. The tracking results videos with this layout are attainable at http://www-iri.upc.es/people/ralqueza/S5_NN.mpg, S5_Bayes.mpg, S8_NN.mpg, S8_Bayes.mpg, S9_NN.mpg and S9_Bayes.mpg.

For comparison purposes, tracking of the target objects in the test sequences was also carried out by applying the six following methods, which only need the ROI window mark in the first frame of the test sequence: Template Match by Correlation, which refers to normalized correlation template matching [9]; Basic Meanshift [10]; Histogram Ratio Shift [11]; Variance Ratio Feature Shift [12]; Peak Difference Feature Shift [12]; and Graph-Cut Based Tracker [13].

From the tracking results of all the tested methods, two evaluation metrics were computed for each frame: the **spatial overlap** and the **centroid distance** [14]. The spatial overlap is defined as the overlapping level $A(GT_k,ST_k)$ between the ground truth GT_k and the system track ST_k in a specific frame k:

$$A(GT_k, ST_k) = \frac{\text{Area}(GT_k \cap ST_k)}{\text{Area}(GT_k \cup ST_k)} \tag{4}$$

and *Dist*(*GTC*$_k$, *STC*$_k$) refers to the Euclidean distance between the centroids of the ground truth (*GTC*$_k$) and the system track (*STC*$_k$) in frame *k*. Naturally, the larger the overlap and the smaller the distance, the better performance of the system track.

Since the centroid distance can only be computed if both GT_k and ST_k are non-null, a **failure ratio** was measured as the number of frames in which either GT_k or ST_k was null (but not both) divided by the total number of frames. Finally, an **accuracy** measure was computed as the number of good matches divided by the total number of frames, where a good match is either a true negative or a true positive with a spatial overlap above a threshold of 0.243 (which is the overlap obtained between two circles of the same size when one of the centers is located in the border of the other circle).

Tables 1, 2 and 3 present the results (mean ± std. deviation) of the two former evaluation measures together with the failure ratio and accuracy of each tracking method for the three tests (best values in bold). Our PIORT tracking methods worked fine in the three test sequences, obtaining the best values of the evaluation measures and outperforming clearly the rest of the methods compared.

Table 1. Tracking performance results on blue ball test sequence (103 frames)

Tracking method	Spatial Overlap	Centroid Distance	Failure Ratio	Accuracy
1Template Match by Correlation	0.275 ± 0.481	74.65 ± 91.53	0.192	0.433
2 Basic Meanshift	0.234 ± 0.523	78.40 ± 90.33	0.192	0.365
3 Histogram Ratio Shift	0.155 ± 0.450	125.88 ± 111.80	0.433	0.298
4 Variance Ratio Feature Shift	0.197 ± 0.375	96.72 ± 134.84	0.385	0.596
5 Peak Difference Feature Shift	0.281 ± 0.566	103.60 ± 136.77	0.413	0.587
6 Graph-Cut Based Tracker	0.007 ± 0.287	188.79 ± 118.13	0.750	0.212
7 Our Tracker PIORT-Neural Net	**0.603 ± 0.400**	12.53 ± 59.38	**0.048**	**0.952**
8 Our Tracker PIORT-Bayesian	0.586 ± 0.394	**12.46 ± 59.40**	**0.048**	**0.952**

Table 2. Tracking performance results on pedestrian test sequence (215 frames)

Tracking method	Spatial Overlap	Centroid Distance	Failure Ratio	Accuracy
1Template Match by Correlation	0.441 ± 0.307	25.25 ± 61.10	0.066	0.772
2 Basic Meanshift	0.241 ± 0.581	72.08 ± 64.33	0.066	0.336
3 Histogram Ratio Shift	0.354 ± 0.237	13.49 ± 38.27	**0.024**	0.644
4 Variance Ratio Feature Shift	0.453 ± 0.320	34.27 ± 81.13	0.118	0.820
5 Peak Difference Feature Shift	0.503 ± 0.203	11.42 ± 45.11	0.033	0.953
6 Graph-Cut Based Tracker	0.039 ± 0.323	194.7 ± 105.3	0.772	0.161
7 Our Tracker PIORT-Neural Net	**0.790 ± 0.238**	11.90 ± 50.87	0.043	**0.957**
8 Our Tracker PIORT-Bayesian	0.737 ± 0.244	**11.15 ± 48.14**	0.038	0.953

Table 3. Tracking performance results on guy-on-Segway test sequence (297 frames)

Tracking method	**Spatial Overlap**	**Centroid Distance**	**Failure Ratio**	**Accuracy**
1Template Match by Correlation	0.102 ± 0.526	130.3 ± 69.75	**0.003**	0.149
2 Basic Meanshift	0.221 ± 0.126	41.30 ± 58.70	0.010	0.402
3 Histogram Ratio Shift	0.527 ± 0.252	22.83 ± 58.43	0.054	0.861
4 Variance Ratio Feature Shift	0.691 ± 0.249	27.69 ± 75.15	0.101	0.895
5 Peak Difference Feature Shift	0.556 ± 0.207	29.19 ± 74.65	0.101	0.895
6 Graph-Cut Based Tracker	0.136 ± 0.218	101.6 ± 112.7	0.365	0.193
7 Our Tracker PIORT-Neural Net	0.734 ± 0.156	**3.40 ± 14.77**	**0.003**	0.973
8 Our Tracker PIORT-Bayesian	**0.743 ± 0.132**	3.70 ± 14.61	**0.003**	**0.980**

5 Conclusions and Future Work

In this paper, we have compared two static recognition methods which are embedded in a probabilistic framework for object recognition and tracking in video sequences called PIORT. Both methods are based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The first classifier is based on a maximum likelihood Bayesian method in which the conditional probabilities for object classes are obtained from the information of the class histograms (for discretized RGB values) and a uniform conditional probability is assumed for the background. The second classifier is based on a neural net which is trained with the RGB colour averages extracted for each spot of the segmented images.

Even though the characteristics of these two classifiers are quite different, the recognition and tracking results of PIORT using both approaches were similar in the three test sequences, which means that the good ability of PIORT to track the objects is mostly due to a smart cooperation of the three inner modules and is not very dependent on the specific method used for object recognition. In the experimental comparison with other reported methods for object tracking, PIORT obtained the best results and much better in most of the cases than those of the other methods. However, as observed in some frames of the test sequences, still there are cases where the behaviour of the tracking decision module of PIORT should be improved, particularly in the step of object re-emergence after occlusion and when other objects of similar appearance are next to the target. The upgrade of this tracking module will be subject of future research.

Acknowledgements

This research was partially supported by Consolider Ingenio 2010, project CSD2007-00018, by the CICYT project DPI 2007-61452 and by the Universitat Rovira I Virgili (URV) through a predoctoral research grant.

References

1. Senior, A., et al.: Appearance models for occlusion handling. J. Image Vis. Comput. 24(11), 1233–1243 (2006)
2. Nguyen, H.T., Smeulders, A.W.M.: Fast occluded object tracking by a robust appearance filter. IEEE Trans. Pattern Anal. Mach. Intell. 26(8), 1099–1104 (2004)
3. Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. 13(11), 1491–1506 (2004)
4. Ito, K., Sakane, S.: Robust view-based visual tracking with detection of occlusions. In: Int. Conf. Robotics Automation, vol. 2, pp. 1207–1213 (2001)
5. Hariharakrishnan, K., Schonfeld, D.: Fast object tracking using adaptive block matching. IEEE Trans. Multimedia 7(5), 853–859 (2005)
6. Amézquita Gómez, N., Alquézar, R., Serratosa, F.: Dealing with occlusion in a probabilistic object tracking method. In: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, Alaska (2008)
7. Alquézar, R., Amézquita Gómez, N., Serratosa, F.: Tracking deformable objects and dealing with same class object occlusion. In: Fourth Int. Conf. on Computer Vision Theory and Applications (VISAPP 2009), Lisboa, Portugal (2009)
8. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
9. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Machine Intell. 25(4), 564–577 (2003)
10. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Machine Intell. 24(5), 603–619 (2002)
11. Collins, R., Zhou, X., The, S.K.: An open source tracking testbed and evaluation web site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2005 (2005)
12. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. IEEE Trans. Pattern Anal. Machine Intell. 27(10), 1631–1643 (2005)
13. Bugeau, A., Pérez, P.: Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. In: Third Int. Conf. on Computer Vision Theory and Applications (VISAPP 2008), Funchal, Madeira, Portugal (2008)
14. Yin, F., Makris, D., Velastin, S.A.: Performance evaluation of object tracking algorithms. In: 10th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2007 (2007)

Real-Time Stereo Matching Using Memory-Efficient Belief Propagation for High-Definition 3D Tele-Presence Systems*

Jesús M. Pérez[1], Pablo Sánchez[1], and Marcos Martinez[2]

[1] University of Cantabria, Av. Los Castros S/N, 39005 Santander, Spain
{chuchi,sanchez}@teisa.unican.es
[2] DS2, Charles Robert Darwin 2, Parc Tecnológic, 46980, Paterna, Valencia, Spain
marcos.martinez@ds2.es

Abstract. High-definition 3D video is one of the features that the next generation of telecommunication systems is exploring. Real-time requirements limit the execution time of stereo-vision techniques to 40-60 milliseconds. Classical belief propagation algorithms (BP) generate high quality depth maps. However, the huge number of required memory accesses limits their applicability in real systems.

This paper proposes a real-time (latency inferior to 40 millisenconds) high-definition (1280x720) stereo matching algorithm using belief propagation. There are two main contributions. The first one is an improved BP algorithm with occlusion, potential errors and texture-less handling that outperforms classical multi-grid bipartite-graph BP while reducing the number of memory accesses. The second one is an adaptive message compression technique with low performance penalty that greatly reduces the memory traffic. The combination of these techniques outperforms classical BP by about 6.0% while reducing the memory traffic by more than 90%.

Keywords: Stereo-vision, Belief propagation, High-Definition, Real-Time, FPGA.

1 Introduction

In current telecommunication systems, the participants normally do not have the feeling of being physically together in one place. In order to improve the immersive face-to-face experience, tele-presence systems are starting to include 3D video and depth estimation capabilities. A typical requirement for these systems [1] includes high definition (at least 1280x720 pixels), good immersive feeling (more than 80 disparity levels) and low latency (depth estimation in less than 40 milliseconds).

Stereo matching using Belief Propagation (BP) is one of the most effective depth estimation techniques, covering the first positions in the Middlebury rankings. Most

* This work has been partially supported by the Spanish CDTI under project CENIT-VISION 2007-1007 and the CICYT under TEC2008-04107.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 825–833, 2009.

of the work using BP is based on the global approach presented in [3], because it converges faster and reduces the memory requirements. However, the execution time of this algorithm in a CPU cannot satisfy real-time (RT) requirements with high-definition (HD) images. Other works [4] are focused on local or semi-global methods. They reduce the execution time, but they normally lose performance. There are some BP algorithms that have been implemented in GPUs although they have limited performance, working with low-resolution images and a small number of disparity levels [5][6]. Finally, several FPGA-based implementations of BP algorithms have been proposed. In [7], an approach that works with low-resolution images and 16 depth levels is proposed. In [2], a RT architecture is presented. However, they work with only 16 disparity levels and a phase-based depth estimation algorithm, which performs worse than BP-based algorithms. A recent publication on FPGA [20] is also focused on implementing BP-based stereo matching on RT. However, our proposal outperforms [20] in three key aspects: first, it perform 1843.2 million disparity estimations per second (obtained as "width*height*fps*Disparity_labels"), being three time faster than [20]. Secondly, their results are close to the BP-M algorithm, which shows poorer results than our proposal. Finally, our proposal can be implemented in a FPGA, while the one in [20] is an ASIC (very expensive and ad-hoc solution).

With recent hardware advances, memory bandwidth has become a more performance-limiting factor than the total number of algorithm operations. To confront this problem, the image is split into several unconnected regions in [8]. The main drawback for RT applications is that the size of the regions is normally very small and this greatly reduces performance.

Here, we present a stereo matching algorithm based on BP. It includes occlusion, potential error and texture-less region handling. Several techniques have been used in stereo matching for occlusion handling [9]. A simple method of detecting occlusion is the cross-checking technique [10]. Other occlusion-handling approaches generate better results [11] but they double the computational complexity. Some other techniques have improved depth estimation in texture-less areas [12]. However, they work with low-resolution images, 48 disparity levels and they do not satisfy RT requirements. Other approaches try to reduce potential error [13], but they work with medium-resolution images, 14.7fps and 40 disparity levels.

In this paper, we propose a global approach based on a double serial BP. A recently presented work [14] also uses a two-step depth estimation algorithm, although with a local approach. Moreover, it does not comply with RT and HD requirements. Some proposals [15] use several BP modules and show better performance than ours. However, the time they spend to obtain a small image disparity map is 250 times the time we use to obtain a HD disparity map. On the other hand, some proposals have concentrated on reducing the number of messages in the BP [16][17] or on compressing the messages to reduce memory [18]. However, they are not able to meet HD, RT constraints and good results.

The system described here presents a BP architecture that complies with actual tele-presence system requirements [1]. The proposal includes two main contributions:

1. It splits the algorithm into two BPs that work serially. Between the two blocks, a new data-cost is calculated based on a pixel classification. This classification identifies occlusions, potential-error, texture-less and reliable pixels. This contribution

improves the single BP results while reducing the number of memory accesses for HD and RT systems (250 times faster than [15]).

2. It defines an adaptive message compression technique to reduce memory traffic with little performance penalty. It provides better balance between performance, simplicity and implementation than [16][17][18] . Moreover, [18] shows some limitations: the message compression used in [18] is not lineal, which means it has to uncompress, operate and compress again. In contrast, our proposal operates with compressed messages. Moreover, as was pointed out in [20], in [18] they assume data to be stored with floating point precision, but if the data precision is 8-bit, only 30-50% compression rate can be achieved. Our proposal achieves more than 70%.

The remainder of this paper is organized as follows. In Section 2, we comment the requirements of the tele-presence system. In Section 3, we discuss the double BP with occlusion, error and texture-less handling methods, as well as the compression technique used to meet the memory access requirements. Finally, we present the experimental results and conclusions in Section 4 and 5.

2 System Requirements

The tele-presence system, which is developed in [1] must satisfy the following constrains:

1. Real-time system with low latency: the depth-estimation processing time is limited to about 40 milliseconds. This requirement is essential to provide presence feeling.
2. High resolution: the image size is 1280x720 pixels. At this resolution, the cameras have a maximum frame rate of 30 fps.
3. Immersion feeling: in order to obtain a life-like 3D model, at least 80 disparity levels seems to be needed. Additionally, a high-quality depth-estimation algorithm (i.e. Belief Propagation) is necessary.
4. Memory bandwidth of the hardware platform: an actual high-performance platform (for example, a commercial FPGA-based ASIC-prototyping board) has a limited maximum external-memory bandwidth (about 153 Gb/seg in the case of the paper reference platform [19]).

As far as the authors know, there are no previous works that can satisfy all these requirements.

3 Proposed System Architecture

In order to use reference [3]'s algorithm in a real system, several parameters have to be defined. In this work we have assumed that the minimum number of iterations and levels needed to cover section 2 requirements is 7. The algorithm variables are quantified using 16 bits and the number of disparity levels is set to 80. The linear truncated model [3] was chosen for the messages as it presents a good balance between edge information and noise information. With these parameters the BP-based technique presented in [3] satisfies the quality constraint (point 3) in the previous section, despite edge error, occlusions and texture-less region flaws. However, it cannot satisfy the RT and memory bandwidth restrictions (points 1 and 4). This algorithm will be referred to as classical BP.

One the most restrictive parameters is the number of external memory accesses. The actual high-performance platform, which is used as the hardware reference model in this work [19], could support up to 6 DDR2-400 memories with 64 bits per memory data bus. The maximum number of memory accesses that a depth estimation algorithm can perform in this platform is about 384 million. Two parameters have been taken into account to obtain this limit: the algorithm variables are quantified with 16 bits and the estimation time is less than 40 milliseconds. However, if the classical BP algorithm is analyzed with the section 3 parameters, the total number of required memory accesses will be 2881 million. Thus, the system is far from being implementable in an actual high-performance platform and it would require a reduction in the number of accesses by almost 90%.

In order to handle occlusions, potential-errors and texture-less regions that degrade the performance of the classical approach, the proposal is to split the BP algorithm into two separate BP blocks. Between them, a new module (Occlusion, Error and texture-less handling module, OE) classifies the pixels into four categories. Additionally, this module will recalculate the values of the cost function taking into account the pixel category. Hereinafter, this algorithm will be denoted as Real-time High-Definition Belief Propagation (RT-HD BP). It performs the following steps:

1. Read left and right images and **compute data-cost**
2. Iterative **BP** (BP1) **over all the pixels**
3. **Output**: for each pixel, send to the output:
   a) **Minimum disparity label of the left-image** depth map.
   b) **Third minimum disparity label of the left-image** depth map.
   c) **Minimum disparity label of the right-image** depth map.
4. **Classify pixels** into reliable, occlusion, error and texture-less (OE Module)
5. Calculate **new data-cost** based on previous classification (OE Module)
6. Iterative **BP** (BP2) only **over non-reliable pixels**
7. **Output**: for each pixel, send to the output:
   - **Minimum disparity label of the left** depth map (final result).

The aim of BP1 is to provide the OE module with enough information to classify the image pixels. A very important advantage of the proposed technique is that this classification can be obtained with a relatively low number of iterations. After the pixel classification has been obtained, the second BP (BP2) generates the final depth map with a reduced number of iterations. Moreover, it also saves memory traffic, performing message passing only on non-reliable pixels (about 20% of the pixels).

It might seem that the complexity and memory bandwidth requirements of the proposed technique could double the classical BP (there are 2 BP blocks, steps 2 and 6). However, the BP1 and BP2 blocks can be implemented in the same hardware module, as they have exactly the same architecture. Moreover, the total number of memory accesses is reduced with respect to classical BP. In table 1, the number of iterations for each level in RT-HD and classical BP are presented. In classical BP, the number of iterations is constant, but in RT-HD BP it changes with the level. Table 1 presents the total number of BP1 and BP2 iterations per level.

Even though the number of iterations is higher in the first levels (6 to 3), the algorithm reduces the iterations in the last level and this minimizes the total number of memory accesses: the classical BP algorithm needs 9.33x accesses while the RT-HD BD needs only 7.47x accesses (19.89% less memory traffic). The parameter 'x' is a function of the image size and disparity levels.

Table 1. Relation between memory accesses and levels

Level	0	1	2	3	4	5	6
Classical BP iterations	7	7	7	7	7	7	7
RT-HD BP iterations	5	7	7	14	14	14	14
Memory accesses per iteration	x	x/4	x/16	x/64	x/264	x/1024	x/4096

The reduction of the number of iterations in the most computationally expensive step is a consequence of two advantages of the proposal. First of all, BP1 makes use of an empirical observation: most of the pixels that converge to correct values will normally do it in a low number of iterations. Thus, the number of iterations of the BP1 block can be very little. Secondly, after the pixel classification, the pixel data cost depends of the pixel type and this improves BP2 convergence. Additionally, BP2 only performs message passing over non-reliable pixels, reducing the number of iterations. Both contributions reduce the number of iterations and memory accesses but their computational impact is very limited.

Occlusion, Edge Error and Texture–Less Area Handling

In the RT-HD BP algorithm, the pixels are classified in 4 categories in the OE module: occluded, potential error, texture-less and reliable pixels.

The OE module generates the occlusion map using a cross-checking technique based on [10]. The module also detects low-textured areas by observing differences between the first ten minimum values on the fly. When the medium difference is bellow an experimental constant, the pixel is classify as texture-less.

In BP, the disparity value for a given pixel is the label index that minimizes the sum of incoming messages and data-cost. When a pixel has converged in the BP algorithm, the sum of the incoming belief messages (SoIM function) tends to have a linear "V" shape (Figure 1.a). This shape is centered on the label index (disparity value). It has been empirically observed that the pixels that converge will normally present a SoIM function with a well defined "V" shape during the first iterations of the last levels (0, 1) in the BP1 module, while the rest of the pixels normally present a non-"V" shape or a SoIM function with several local minima (Figure 1.b).

Based on this observation, the proposed algorithm includes a simple technique to identify the pixels that probably converge. It is based on the comparison between the disparity label of the first and the third minimum. If the SoIM function has a "V" shape, the first (1M in figure 1), second (2M) and third (3M) minimum disparity values will normally be consecutive values. However, if the shape is different, the third value will not normally be a consecutive value (Figure 1.b). This simple observation normally produces good results with a very low computational effort. The pixel whose SoIM function has a "V" shape will be classified as a reliable pixel and the rest are classified as potential error pixels. As occluded and textured-less pixels have previously been identified, the pixels that are classified as reliable have a high probability of having converged to the correct value.

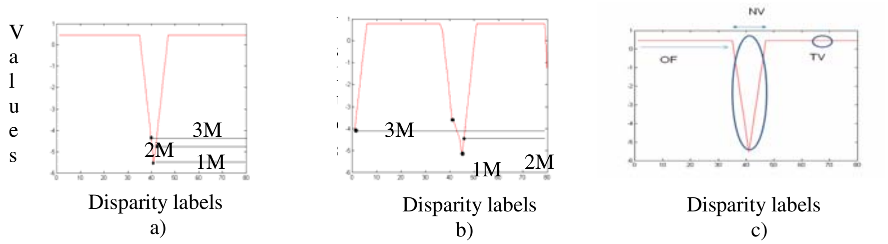
The OE module generates a two-bit per pixel map that classifies the pixels in four categories: reliable, edge-error, occlusion and texture-less pixels.

New Data Cost

This module uses the information provided by OE to calculate new data costs as:

1. Reliable pixels: data cost defined as 0 for their minimum disparity label and a predefined penalty, equal to the maximum truncated value, for the rest labels.
2. Texture-less pixels: The data cost is 0 for all the disparity labels (unknown). This helps texture-less pixels to obtain correct disparity values.
3. Error pixels: they keep their data cost.
4. Occluded pixels: take the value of the first non-occluded pixel on their left.

BP2 limits the message passing to non-reliable pixels, reducing the memory traffic. The total memory reduction is about 21%. This reduction is still far from the required 90%. To reach this limit, a new improvement has been developed.



Disparity labels
a)

Disparity labels
b)

Disparity labels
c)

Fig. 1. a)Reliable pixel b) Possible error pixel c) Parameters for message compression

Adaptive Message Compression

The proposed compression method is based on the shape of the belief messages. Instead of storing only the envelope points, as in the EPG method in [18], the proposed technique stores all the points inside a region around the minimum disparity label. It has two main advantages with respect to [18]. Firstly, we do not need to uncompress the message prior to operate with it. Secondly, the compression rate drastically decreases when using EPG with limited precision. In contrast, we achieve more than 70% even with fixed point variables.

In our proposal, the number of stored points is a function of several parameters (adaptive approach): iteration, level and pixel type. This can reduce the compression factor, but increase the performance and reduce the quality penalty. This adaptive technique is applied only to the BP2 block reducing the memory traffic by about 70%. The proposed compression technique stores 3 parameters and a set of points per message (Figure 1c):

1. The offset (OF): first disparity label of the selected region.
2. Number of disparity labels (NV) of the selected region.
3. Information values: all the values of the selected region (from OF to OF+NV).
4. Truncating value (TV): value assigned to all the disparity labels outside the NV.

The pixels that converge will normally present a shape that is easily and efficiently compressed with the proposed techniques. This property, combined with the pixel classification that the OE generates, guarantees a good compression factor.

4 Results

In order to validate the proposed algorithms, several video sequences [21] have been evaluated with the classical and the RT-HD BP. In Fig 2, we show the disparity maps that are obtained with classical BP (a), the proposed RT-HD BP without compression (b) and the RT-HD BP with enough compression to meet the RT restrictions in section 2 (c). Some occlusions have been mitigated, some errors corrected and some texture-less zones have been filled in (b,c).

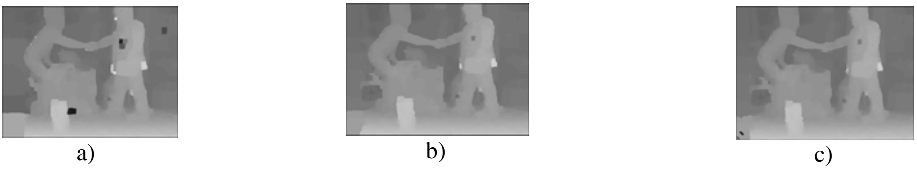
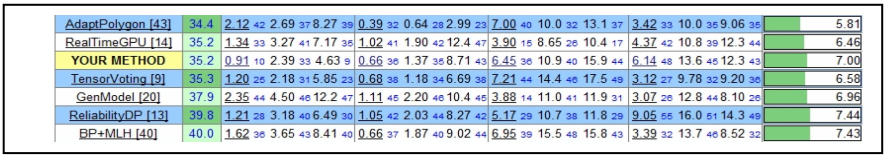


Fig. 2. Disparity maps: a)Classical BP b) and c)RT-HD BP without and with compression

The RT-HD BP shows an improvement of more than 6% when compared to classical BP. At the same time, it satisfies section 2's RT and HD requirements. The memory reduction of the RT-HD BP with compression is about 90%. To finish this results section, we provide Middlebury test results for our proposal in Figure 3 and Table 2, comparing the proposal with RT publications ranking Middlebury test. For convenience, we maintain the names used in Middlebury test in Table 2. As can be derived from Table 2, our proposal is the only ranking in the test that is able to achieve true RT for HD images. The only one whose latency is close to RT (RealtimeBP) works with small images. Moreover, between all of the proposals focus on real time, there is only one whose position in the ranking is significantly better than our proposal (PlaneFitBP) but working at 1fps, which is not RT at all.



AdaptPolygon [43]	34.4	2.12 42	2.69 37	8.27 39	0.39 32	0.64 28	2.99 23	7.00 40	10.0 32	13.1 37	3.42 33	10.0 35	9.06 35	5.81
RealTimeGPU [14]	35.2	1.34 33	3.27 41	7.17 35	1.02 41	1.90 42	12.4 47	3.90 15	8.65 26	10.4 17	4.37 42	10.8 39	12.3 44	6.46
YOUR METHOD	35.2	0.91 10	2.39 33	4.63 9	0.66 36	1.37 35	8.71 43	6.45 36	10.9 40	15.9 44	6.14 48	13.6 45	12.3 43	7.00
TensorVoting [9]	35.3	1.20 25	2.18 31	5.85 23	0.68 38	1.18 34	6.69 38	7.21 44	14.4 46	17.5 49	3.12 27	9.78 32	9.20 36	6.58
GenModel [20]	37.9	2.35 44	4.50 46	12.2 47	1.11 45	2.20 46	10.4 45	3.88 14	11.0 41	11.9 31	3.07 26	12.8 44	8.10 26	6.96
ReliabilityDP [13]	39.8	1.21 28	3.18 40	6.49 30	1.05 42	2.03 44	8.27 42	5.17 29	10.7 38	11.8 29	9.05 55	16.0 51	14.3 49	7.44
BP+MLH [40]	40.0	1.62 36	3.65 43	8.41 40	0.66 37	1.87 40	9.02 44	6.95 39	15.5 48	15.8 43	3.39 32	13.7 46	8.52 32	7.43

Fig. 3. Middlebury ranking for our proposal

Table 2. Proposals focus on real-time, in middlebury ranking, comparison

Parameter	Latency(msec)	Image resolution	Disp. Lev.	Rank.
Proposed RT-HD BP	40	1280x720	80	35.2
RealTime GPU	183	640x480	48	35.2
RTCensus	--	--	--	45.6
Realtime BP	62.5*	320x240	16	30.5
FastAggreg	600	450x675	60	32.7
PlaneFit BP	1000*	512x384	48	12.8

*Latency has been extrapolated from the fps data (18fps≈62.5millisecond and 1fps≈1000 milliseconds).

5 Conclusions

In this work we have presented a Real-Time High-Definition depth estimation algorithm based on Belief Propagation. It estimates depth maps in less than 40 milliseconds for HD images (1280x720 pixels at 30fps) with 80 disparity values. The work exploits the proposed double BP topology and it handles occlusions, potential errors and texture-less regions to improve the overall performance by more than a 6% (compared with classical BP) while it reduces the memory traffic by about 21%. Moreover, the adaptive message compression method allows the system to satisfy Section-2's real-time and low execution latency requirements, reducing the number of memory accesses by more than a 70% with an almost negligible loss of performance (less than 0.5%). The total memory traffic reduction is about 90% with a 6.0% performance improvement (compared with classical BP).

References

1. Vision project (2009), `http://vision.tid.es`
2. Diaz, J., Ros, E., Carrillo, R., Prieto, A.: Real-Time System for High-Image Resolution Disparity Estimation. In: IEEE TIP 2007 (2007)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. In: IEEE CVPR 2004 (2004)
4. Hirschmuller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. In: IEEE TPAMI 2008 (2008)
5. Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M., Nister, D.: Real-time Global Stereo Matching Using Hierarchical Belief Propagation. In: BMCV 2006 (2006)
6. Wang, L., Liao, M., Gong, M., Yang, R., Nister, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: IEEE 3DPVT 2006 (2006)
7. Park, S., Jeong, H.: A fast and parallel belief computation structure for stereo matching. In: IASTED IMSA 2007 (2007)
8. Tseng, Y., Chang, N., Chang, T.: Low Memory Cost Block-Based Belief Propagation for Stereo Correspondence. In: IEEE ICME 2007 (2007)
9. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: IEEE ICCV 2001 (2001)
10. Egnal, G., Wildes, R.: Detecting binocular half occlusions: empirical comparisons of five approaches. In: IEEE PAMI 2002 (2002)

11. Sun, J., Li, Y., Kang, S., Shum, H.: Symmetric stereo matching for occlusion handling. In: IEEE CVPR 2005 (2005)
12. Yang, Q., Engels, C., Akbarzadeh, A.: Near Real-time Stereo for Weakly-Textured Scenes. In: BMCV 2008 (2008)
13. Gong, M., Yang, R.: Image-gradient-guided real-time stereo on graphics hardware. In: IEEE 3DIM 2005 (2005)
14. Yilei, Z., Minglun, G., Yee-Hong, Y.: Local stereo matching with 3D adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy. In: IEEE ICPR 2008 (2008)
15. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling. In: IEEE TPAML 2009 (2009)
16. Huq, S., Koschan, A., Abidi, B., Abidi, M.: Efficient BP Stereo with Automatic Parameter Estimation. In: IEEE ICIP 2008 (2008)
17. Sarkis, M., Diepold, M., Klaus: Sparse stereo matching using belief propagation. In: IEEE ICIP 2008 (2008)
18. Yu, T., Lin Super, R., Bei Tang, B.: Efficient Message Representations for Belief Propagation. In: IEEE ICCV 2007 (2007)
19. http://www.synplicty.com/products/haps/haps-52.html (2009)
20. Liang, C., Cheng, C., Lai, Y., Chen, L., Chen, H.: Hardware-Efficient Belief Propagation. In: IEEE CVPR 2009 (2009)
21. Feldmann, M., Mueller, F., Zilly, R., Tanger, K., Mueller, A., Smolic, P., Kauff, T.: Wiegand HHI Test Material for 3D Video, MPEG 2008/M15413, Archans, Framce (April 2008)

Improving Recurrent CSVM Performance for Robot Navigation on Discrete Labyrinths

Nancy Arana-Daniel[1], Carlos López-Franco[1], and Eduardo Bayro-Corrochano[2]

[1] Electronics and Computer Science Division, Exact Sciences and Engineering Campus, CUCEI, Universidad de Guadalajara, Av. Revolucion 1500, Col. Olímpica, C.P. 44430, Guadalajara, Jalisco, México
[2] Cinvestav del IPN, Department of Electrical Engineering and Computer Science, Zapopan, Jalisco, México
{nancy.arana,carlos.lopez}@cucei.udg.mx,edb@gdl.cinvestav.mx

Abstract. This paper presents an improvement of a recurrent learning system called LSTM-CSVM (introduced in [1]) for robot navigation applications, this approach is used to deal with some of the main issues addressed in the research area: the problem of navigation on large domains, partial observability, limited number of learning experiences and slow learning of optimal policies. The advantages of this new version of LSTM-CSVM system, are that it can find optimal paths through mazes and it reduces the number of generations to evolve the system to find the optimal navigation policy, therefore either the training time of the system is reduced. This is done by adding an heuristic methodoly to find the optimal path from start state to the goal state.can contain information about the whole environment or just partial information about it.

Keywords: Robot navigation, LSTM-CSVM, optimal path, heuristic.

1 Introduction

This paper presents an improvement of the recurrent learning system LSTM-CSVM [1] for robot navigation applications. The design of LSTM-CSVM system is based on Evoke algorithm [evoke], both of them are constructed by two cascaded modules: (1) a recurrent neural network Long-Short Term Memory (LSTM) that receives the sequence of external inputs, and (2) a parametric function that maps the internal activations of the first module to a set of outputs. The second module is different for these systems: Evoke algorithm uses a real Support Vector Machine to produce precise final outputs, meanwhile LSTM-CSVM uses a Clifford generalization of SVM algorithm known as Clifford Support Multivector Machine (CSVM) [3] to do the same job, but improving the real-computation time. So, the main advantage of using CSVM approach as second module is that real-SVM algorithm is transformed into a MIMO SVM without adding complexity. This is done by embedding the optimization problem into a geometric algebra framework, which allows us to represent the entries to the CSVM, the optimization variables and the outputs as multivectors and, in this way we can

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 834–842, 2009.

represent multiple classes according to the dimension of the geometric algebra in which we work.

In previous work [1] it is proved that LSTM-CSVM approach gets lower training and testing errors by showing different experiment results on applications like time series and comparing them with the results obtained from algorithms such as Evolino, Echo State Networks and LSTM. Furthermore, it is shown the performance of LSTM-CSVM on tasks like robot navigation through a maze using reinforcement learning and neuroevolution approaches to solve this tasks.

The results presented in this paper provide evidence about the improvement of the performance of LSTM-CSVM for robot navigation, in particular for maze navigation by adding an heuristic approach which allows us to find the lowest cost path from the robots start state to the goal state. Other advantage of this new version is that the number of generations to evolve the system is reduced, therefore either the training time of the system is reduced.

2 Long Short Term Memory (LSTM)

Learning to extract and represent information from a long time ago has proven difficult, both for model-based and for model-free approaches. The difficulty lies in discovering the correlation between a piece of information and the moment at which this information becomes relevant at a later time, given the distracting observations and actions between them [4].

LSTM is a recurrent neural network architecture, originally designed for supervised timeseries learning [5]. It is based on an analysis of the problems that conventional recurrent neural networks and their corresponding learning algorithms, e.g. Elman networks with standard one step backpropagation, Elman networks with backpropagation through time (BPTT) and real-time recurrent learning (RTRL), have when learning timeseries with long-term dependencies. These problems boil down to the problem that errors propagated back in time tend to either vanish or blow up (see [4]). LSTM's solution to this problem is to enforce constant error ow in a number of specialized units, called Constant Error Carrousels (CECs). This turns out to correspond to the CECs having linear activation functions which do not decay over time. In order to prevent the CECs from filling up with useless information from the timeseries, access to them is regulated using other specialized, multiplicative units, called input gates. Like the CECs, the input gates receive input from the timeseries and the other units in the network, and they learn to open and close access to the CECs at appropriate moments. Access from the activations of the CECs to the output units (and possibly other units) of the network is regulated using multiplicative output gates. Similar to the input gates, the output gates learn when the time is right to send the information stored in the CECs to the output side of the network. The forget gates learn to reset the activation of the CECs (in a possibly gradual way) when the information stored in the CECs is no longer useful. The combination of a CEC with its associated input, output, and forget gate is called a memory cell.

3 Clifford Support Multivector Machine (CSVM)

For the case of the Clifford SVM for classification we represent the data set in a certain Clifford Algebra [6] $\mathcal{G}_n$ where $n = p + q + r$, where any multivector base squares to 0, 1 or -1 depending if they belong to p, q, or r multivector bases respectively. We consider the general case of an input comprising D multivectors, and one multivector output, i.e. each ith-vector has D multivector entries $\boldsymbol{x_i} = [\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, ..., \boldsymbol{x_{iD}}]^{\boldsymbol{T}}$, where $\boldsymbol{x_{ij}} \in \boldsymbol{\mathcal{G}_n}$ and D is its dimension. Thus the ith-vector dimension is D×2^n, then each data ith-vector $\boldsymbol{x_i} \in \boldsymbol{\mathcal{G}_n^D}$. And each of ith-vectors will be associated with one output of the 2^n posibilities given by the multivector output $y_i = y_{i_s} + y_{i_{\sigma_1}} + y_{i_{\sigma_2}} + ... + y_{i_I} \in \{\pm 1 \pm \sigma_1 \pm \sigma_2 \ldots \pm I\}$ where the first subindex s stands for scalar part. The dual expression of the optimization problem to solve by the CSVM[1] is: ,

$$\begin{aligned} & max \quad \boldsymbol{a^T 1} - \frac{1}{2}\boldsymbol{a^T H a} \\ subject\,to & \\ & 0 \leq (\alpha_s)_j \leq C, \; 0 \leq (\alpha_{\sigma_1})_j \leq C, ..., \\ & 0 \leq (\alpha_{\sigma_1 \sigma_2})_j \leq C, ..., 0 \leq (\alpha_I)_j \leq C \\ & for \quad j = 1, ..., l, \end{aligned} \tag{1}$$

where H represents the Gramm matrix and it is defined by the Clifford product of the input vectors x in terms of the matrices of t-grade $\boldsymbol{H_t} = \langle \boldsymbol{x}^{\dagger} \boldsymbol{x} \rangle_{\boldsymbol{t}}$, and $\boldsymbol{a}$, has the dimensions $(l \times 2^n) \times 1$, l is the total number of training data, 2^n is the dimension of the geometric algebra and each entry for the vector is given by:

$$\begin{aligned} \boldsymbol{a_s^T} &= [(\alpha_s)_1 (y_s)_1, \, (\alpha_s)_2 (y_s)_1, \, ..., \, (\alpha_s)_l (y_s)_l] \\ \boldsymbol{a_{\sigma_1}^T} &= [(\alpha_{\sigma_1})_1 (y_{\sigma_1})_1, \, (\alpha_{\sigma_1})_1 (y_{\sigma_1})_1, \, ..., \, (\alpha_{\sigma_1})_l (y_{\sigma_1})_l] \\ ..., & \\ \boldsymbol{a_I^T} &= [(\alpha_I)_1 (y_I)_1, \, (\alpha_I)_1 (y_I)_1, \, ..., \, (\alpha_I)_l (y_I)_l] \end{aligned} \tag{2}$$

where $(\alpha_s)_j, (\alpha_{\sigma_1}), ..., (\alpha_{\sigma_1 \sigma_2})_j, ..., (\alpha_I)_j \leq$ are the Langrange multipliers.

The threshold $\boldsymbol{b} \in \mathcal{G}_n^D$ can be computed by using KKT conditions with the Clifford support vectors as follows $\boldsymbol{b} = (b_s + b_{\sigma_1}\sigma_1 + ... + b_{\sigma_1\sigma_2}\sigma_1\sigma_2 + ... + b_I I)$ $= \sum_{j=1}^{l} \boldsymbol{(y_j - w^{\dagger T} x_j)}/l$.

The decision function can be seen as sectors reserved for each involved class, i.e. in the case of complex numbers ($\mathcal{G}_{1,0,0}$) or quaternions ($\mathcal{G}_{0,2,0}$) we can see that the circle or the sphere are divide by means spherical vectors. Thus the decision function can be envisaged as

$$\boldsymbol{y} = csign_m\Big[f(\boldsymbol{x})\Big] = csign_m\Big[\boldsymbol{w^{\dagger T} x + b}\ \Big] = csign_m\Big[\sum_{j=1}^{l} (\alpha_j \circ \boldsymbol{y_j})(\boldsymbol{x_j^{\dagger T} x}) + \boldsymbol{b}\Big] \tag{3}$$

where $csign_m\Big[f(\boldsymbol{x})\Big]$ is the function for detecting the sign of $f(\boldsymbol{x})$ and m stands for the different values which indicate the state valency, e.g. bivalent, tetravalent and the operation "∘" is defined as

[1] The reader can get a detailed explanation about computations of CSVM in [3].

$$(\alpha_j \circ \boldsymbol{y_j}) =< \alpha_j >_0 < \boldsymbol{y_j} >_0 + < \alpha_j >_1 < \boldsymbol{y_j} >_1 \sigma_1 + ... + < \alpha_j >_{2^n} < \boldsymbol{y_j} >_{2^n} I \qquad (4)$$

simply one consider as coefficients of the multivector basis the multiplications between the coefficients of blades of same degree. The major advantage of our approach is that we redefine the optimization vector variables as multivectors. This allows us to utilize the components of the multivector output to represent different classes. The amount of achieved class outputs is directly proportional to the dimension of the involved geometric algebra. The key idea to solve multi-class classification in the geometric algebra is to avoid that the multivector elements of different grade get collapsed into a scalar, this can be done thanks to the redefinition of the primal problem involving the Clifford product instead of the inner product of the real approach.

For the nonlinear Clifford valued classification problems we require a Clifford valued kernel $K(\boldsymbol{x}, \boldsymbol{y})$. In general we build a Clifford kernel $K(x_m, x_j)$ by taking the Clifford product between the conjugated of $\boldsymbol{x_m}$ and $\boldsymbol{x_j}$ as follows

$$K(x_m, x_j) = \Phi(\boldsymbol{x_m^\dagger})\Phi(\boldsymbol{x_j}), \qquad (5)$$

note that the kind of conjugation operation † of a multivector depends of the signature of the involved geometric algebra $\mathcal{G}_{p,q,r}$. The results of this paper were obtained using the quaternion-valued Gabor kernel function as follows $\boldsymbol{i} = \sigma_2\sigma_3$, $\boldsymbol{j} = -\sigma_3\sigma_1$, $\boldsymbol{k} = \sigma_1\sigma_2$. The Gaussian window Gabor kernel function reads

$$K(\boldsymbol{x_m}, \boldsymbol{x_n}) = g(\boldsymbol{x_m}, \boldsymbol{x_n}) exp^{-\boldsymbol{i}\mathbf{w}_0^T(\boldsymbol{x_m} - \boldsymbol{x_n})} \qquad (6)$$

where $g(\boldsymbol{x_m}, \boldsymbol{x_n}) = \frac{1}{\sqrt{2\pi}\rho} exp^{-\frac{||\boldsymbol{x_m} - \boldsymbol{x_n}||^2}{2\rho^2}}$ and the variables $\boldsymbol{w_0}$ and $\boldsymbol{x_m} - \boldsymbol{x_n}$ stand for the frequency and space domains respectively. Unlike the Hartley transform or the 2D complex Fourier this kernel function separates nicely the even and odd components of the involved signal

4 LSTM-CSVM

LSTM-CSVM is an Evoke based system [2]: the underlying idea of these systems is that it is needed two cascade modules: a robust module to process short and long-time dependencies (LSTM) and an optimization module to produce precise outputs (CSVM, Moore-Penrose pseudoinverse method, SVM respectively). The LSTM module addresses the disadvantage of having relevant pieces of information outside the history window and also avoids the problem of the "vanishing error" presented by algorithms like Back-Propagation Through Time (BPTT) or Real-Time-Recurrent Learning (RTRL)[2]. Meanwhile CSVM maps the internal activations of the fist module to a set of precise outputs, again, it is taken advantage of the multivector output representation to implement a system with less process units and therefore less computational complex.

[2] The reader can get more information about BPTT and RTRL-vanishing error versus LSTM-constant error flow in [5].

LSTM-CSVM works as follows: a sequence of input vectors $(\boldsymbol{u(0)...u(t)})$ is given to the LSTM which in turn feeds the CSVM with the outputs of each of its memory cells.

The CSVM aimed at finding the expected nonlinear mapping of training data. The input and output equations are:

$$\phi(t) = f(\boldsymbol{W,} \boldsymbol{u}(t), \boldsymbol{u}(\text{t-1}),...,\boldsymbol{u}(0),...,).$$

$$y(t) = b + \sum_{i=1}^{k} w_i K(\phi(t), \phi_i(t)). \tag{7}$$

where $\phi(t) = [\psi_1, \psi_2, ..., \psi_n]^T \in R^n$ is the activation in time t of n units of the LSTM, this serves as input to the CSVM, given the input vectors$(\boldsymbol{u(0)...u(t)})$ and the weight matrix $\boldsymbol{W}$. Since the LSTM is a recurrent net, the argument of the function $f(.)$ represents the history of the input vectors.

In the first phase of the training, data were propagated through the LSTM- module of the system, it was training using reinforcement learning with advantage-λ learning to produce a vector of activations $\phi(t)$. Once all k sequences have been seen, the weights w_{ij} of the CSVM are computed using support vector regression/ classification from ϕ to the desired outputs d_i, with $\{\phi, d_i\}$ as training set.

In the second phase, a validation set is presented to the network, but now the inputs are propagated through the LSTM and the newly computed output connections to produce $y(t)$. The error in the classification/prediction or the residual error, possibly combined with the error on the training set, is then used as the fitness measure to be minimized by evolution. By measuring error on the validation set rather that just the training set, LSTM will receive better fitness for being able to generalize. Those LSTM that are most fit are then selected for reproduction where new candidate LSTM are created by exchanging elements between chromosomes and possibly mutating them. New individuals replace the worst old ones and the cycle repeats until a sufficiently good solution is found. Those LSTM that are most fit are then selected for reproduction where new candidate RNNs are created by exchanging elements between chromosomes and possibly mutating them. New individuals replace the worst old ones and the cycle repeats until a sufficiently good solution is found. We evolved the rows of the LSTM's weight matrix using the evolutionary algorithms known as Enforced Sub-Populations (ESP) [7] algorithm. This approach differs with the standard methods, because instead of evolving the complete set of the net parameters, it allows to evolve subpopulations of the LSTM memory cells. For the mutation of the chromosomes, the ESP uses Cauchy density function.

5 LSTM-CSVM with Cost Heuristic for Robot Navigation through Mazes

We built an LSTM-CSVM system in order to deal with the path planning problem for one robot moving through a maze of obstacles to a goal. The recurrent neural system is used as the function approximator for a model-free, value function-based reinforcement learning algorithm. The state of the environment is approximated by the current observation, (which is the input to the network) together with the recurrent activations in the network, which represent the agent's history. In this case, the recurrent activations in the specialized memory cells, are supposed to learn to represent relevant information from long ago.

The number of input units of the system is stablished by the size of the observation vector, which is four and represents if there is (1) or there is not (0) an obstacle on a particular adjoining position (North, South, East or West). The LSTM module has four memory cells and their activation output values feed one CSVM module which give the final four outputs as a multivector, each element of it represents one of the posible actions to take by the agent on the current state.

We added to the reinforcement system LSTM-CSVM a heuristic approach: on each step t of the learning, we compute a heuristic reinforcement signal $h(t)$. The heuristic reinforcement signal idea is to provide a comparative measure of output action goodnes for each state. These computations are made by a module that we called "'the critic"' [8]. At each time t, the environment provides the path-finder with the input pattern $u(t)$ (environment observation), together with the environmental reinforcement signal $z(t)$. The input pattern is fed to both the LSTM-CSVM and the critic. Nevertheless the LSTM-CSVM does not receive directly the environmental reinforcement signal but the *heuristic reinforcement signal* $h(t)$ elaborated by the critic. The latter is that the LSTM-CSVM produces instantaneously an output patter $y(t)$ that is a multivector which represents the action to execute by the agent. The environment receives this action $y(t)$ and, at time $t+1$, sends to the LSTM-CSVM both an evaluation $z(t+1)$ of the appropriateness of the action $y(t)$ for the observation $u(t)$ and a new stimulus $u(t+1)$.

5.1 The Critic

The goal of the critic is to transform the environmental reinforcement signal into a more informative signal, namely the heuristic reinforcement signal. This improvement is based on past experience of the path-finder when interacting with the environment, as represented bye the reinforced baseline b:

$$h(t) = z(t) - b(t-1) \tag{8}$$

The critic receives the input pattern X(t) and predicts the reinforcement baseline b(t) with which to compare the associated environment reinforcement signal z(t + 1). We use the technique called predicted comparison [9], it computes the reinforcement baseline as a prediction based on the environmental reinforcement received when the sameor similarinput patterns occurred. That is, the critic has to predict the environmental reinforcement signal $z(t+1)$ to be received by the path-finder when the stimulus $u(t)$ is present. In order to undertake this task, the critic is built as a second network with a supervised learning algorithm to learn to associate each stimulus with the corresponding environmental reinforcement

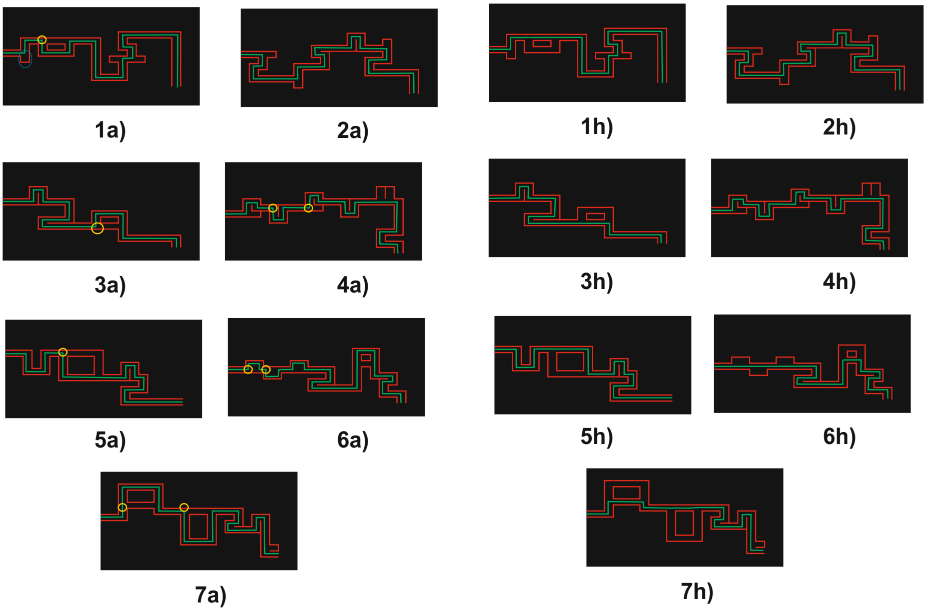


Fig. 1. Left side maze images (1a to 7a) shows path results obtained with LSTM-CSVM without heuristic. Right side shows optimal paths obtained with heuristic approach of LSTM-CSVMv(1h to 7h).

signal. In particular, we have used the "on-line" version of the backpropagation algorithm with a momentum term. the finding path improves, the mapping from stimuli to reinforcement changes.

5.2 Results

We evolved the recurrent LSTM-CSVM system during 30 generations using a Cauchy noise parameter of $\alpha = 10^{-}3$. The first phase of the experiments consisted of simulated mazes discretized on 5 by 5 pixels cells. The length of each maze varies from 100 to 300 cells. Left side of figure 1 shows the paths found with LSTM-CSVM approach (which was evolved during 60 generations using a Cauchy noise parameter of $\alpha = 10^{-}3$) meanwhile right side shows the shortest paths obtained with heuristic approach of LSTM-CSVM. It is important to note that on results obtained with LSTM-CSVM approach the agent seems to have learned a policy which tells the agent to take action go-north every time it finds a two junction like shown on (Fig. 1-3a), 7a)). The policy also tells the agent to take the action go-south on two junction like shown on (Fig. 1-1a), 5a)). These choices of actions produce suboptimal paths.

Phase two of agent navigation through mazes was applied to real mobile robot system, it comprises mobile-base robot, a stereoscopic camera and a laser sensor. The task consisted to move the robot through a real 2D labyrinth. The labyrinths

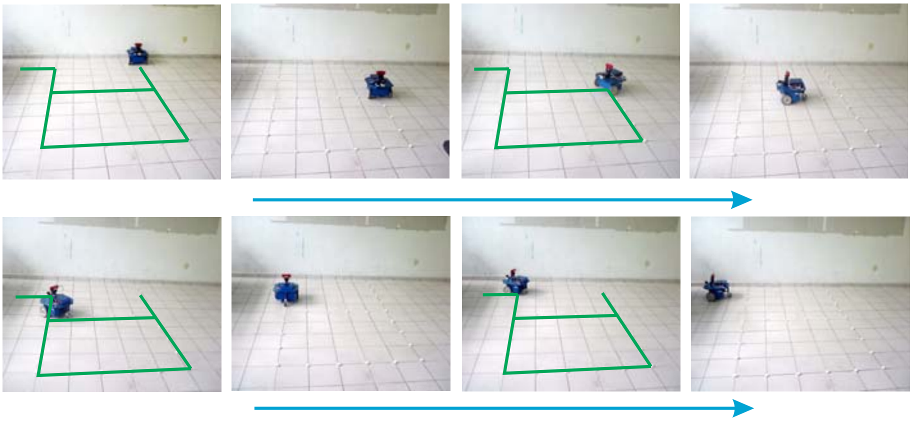


Fig. 2. Image sequence of a mobile robot following the shortest path finding by the heuristic approach of LSTM-CSVM. Blue arrows show the reading order of the images.

used on this phase were much more shorter than the simulated on phase one. The mazes were discretized on 10 by 10 pixels cells and the length of each maze varied from 20 to 30 cells. The LSTM-CSVM with heuristic approach was evolved during 15 generations and the one path found is shown on (Fig. 2). Note that the mobile robot takes the shortest path to the final goal position.

6 Conclusion

In the robot path finding problem , the consequences of an action can emerge later in time. Thus, actions must be selected based on both their short- and long-term consequences. Neverthless many learning algorithms have limited long-term memory. The approach presented in this paper overcomes this limitation thanks to its LSTM module which is responsible to capture the short- and long-term correlation between input data sequences meanwhile the CSVM module allows to produce precise outputs. By adding the heuristic approach to the LSTM-CSVM system we can be able to overcome the problem of time consuming of traditional reinforcement learning in the initial learning phase, this heuristic methodology also allows to find shortest paths as is shown in the results subsection and along the numerous experiments that we had conducted.

References

1. Bayro-Corrochano, E., Arana-Daniel, N., Vallejo-Gutierrez, R.: Recurrent Clifford Support Machines. In: Proceedings IEEE World Congress on Computational Intelligence, Hong-Kong (2008)
2. Schmidhuber, J., Gagliolo, M., Wierstra, D., Gomez, F.: Recurrent Support Vector Machines, Technical Report, no. IDSIA 19-05 (2005)

3. Bayro-Corrochano, E., Arana-Daniel, N., Vallejo-Gutierrez, R.: Geometric Preprocessing, geometric feedforward neural networks and Clifford support vector machines for visual learning. Journal Neurocomputing 67, 54–105 (2005)
4. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient ow in recurrent nets: the difficulty of learning long-term dependencies. In: Kremer, S.C., Kolen, J.F. (eds.) A field guide to dynamical recurrent neural networks. IEEE Press, Los Alamitos (2001)
5. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory, Technical Report FKI-207-95 (1996)
6. Hestenes, D., Li, H., Rockwood, A.: New algebraic tools for classical geometry. In: Sommer, G. (ed.) Geometric Computing with Clifford Algebras. Springer, Heidelberg (2001)
7. Gómez, F.J., Miikkulainen, R.: Active guidance for a finless rocket using neuroevolution. In: Proc. GECCO, pp. 2084–2095 (2003)
8. Millán, J.R., Torras, C.: A Reinforcement Connectionist Approach to Robot Path Finding in Non-Maze-Like Environments. J. Mach. Learn. 8, 363–395 (1992)
9. Sutton, R.S.: Temporal credit assignment in reinforcement learning. Ph.D. Thesis, Dept. of Computer and Information Science, University of Massachusetts, Amherst (1984)

Discrete Integral Sliding Mode Control in Visual Object Tracking Using Differential Kinematics

Luis Enrique González Jiménez, Alexander Loukianov,
and Eduardo Bayro-Corrochano

CINVESTAV, Department of Automatic Control, Unidad Guadalajara,
Av. Científica 1145, 45015 Zapopan, Jalisco, México
{lgonzale,louk,edb}@gdl.cinvestav.mx

Abstract. A Discrete Integral Sliding Mode algorithm is proposed to control a Stereo Vision System (SVS) and perform Visual Object Tracking. The kinematical model of the structure is obtained using Geometric Algebra (GA). The localizing part was done in a real SVS in order to obtain the reference for orientation vector and the application for a Pan Tilt Unit is presented. The algorithm presents a good and robust performance.

Keywords: Integral Sliding Mode Control, Visual Tracking, Geometric Algebra.

1 Introduction

Consider a stereo vision system (SVS) that is mounted on a mechanical device which defines the system's orientation related to a defined base frame. The global task can be divided in two parts: first the system must recognize the target in the scene and extract a vector that will characterize the object's position; then the system must re-define its kinematical structure, so its orientation can be aligned with the target's position. Using a stereo system provides the advantage of depth information, this is fundamental in tasks like grasping and manipulating objects, in addition to visual tracking.

1.1 Target Localization

Model-based algorithms use a pre-obtained model of the object, usually composed of lines. From this model, points of interest are projected on the image plane, and when the correspondences of these points are located and the image, the target is located as well [1]. Global Appearance-based methods segment a region from the image containing the object. Among the algorithms used for feature detection are the Harris corner detector and SIFT [2]. Histogram-based methods use histograms obtained from the target and it is compared with a reference image in the scene. The histogram uses usually color [3] or spatial information.

1.2 Kinematical Control

Several algorithms have been used to control robotic devices such as the classic PID controller, Adaptive Control [4], Neural Control [5] and Sliding Mode Control (SMC)

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 843–850, 2009.

[6]. Among these methods SMC is one of the most effective approaches due to its robustness to matched perturbations, model uncertainties, and it demands a low computational cost. On the other hand, the Integral Sliding Mode Control (ISMC) [7] can guarantee the robustness of the closed-loop system throughout the entire response starting from the initial time instance, and permits to reduce the controller gains in comparison with standard SMC. However, due to a finite sampling rate, continuous-time sliding mode control could be inappropriate for a discrete-time system. Hence, a re-work in the sliding-mode control (SMC) strategy for sampled-data systems was necessary [9]. The equivalent control is used in this work to design a chattering-free discrete ISM controller (DISMC) for a SVS, and a performance of this controller is demonstrated.

2 Problem Formulation

A general scheme for a SVS is presented in Figure 1, where T_p is the target's point, A is the orientation vector for the SVS, O_b is the origin of the base frame and O_c is the origin of the camera frame conveniently attached to the last link of the kinematical device.

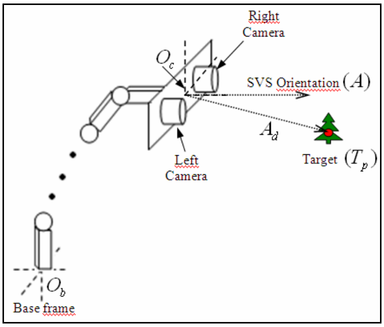


Fig. 1. Stereo Vision System and target

The kinematical model of this structure can be defined as [8]

$$A = f(\theta) \tag{1}$$

where $\theta = [\theta_1, \ldots, \theta_n]^T$ is the angles vector of the system, n is the number of joints in the kinematical structure and the vector $f(\theta)$ is defined by the direct kinematics of the system.

2.1 Kinematical Model with GA

The kinematical model for a serial manipulator can be obtained using a Geometric Algebra approach [10], which has the advantage of being a simple procedure. In this work geometric algebra $G_{3,0,0}$ with the orthonormal base $\{e_1, e_2, e_3\}$ is used.

First we need to define the rotor (geometric entity that defines rotations) for each joint of the system as

$$R_i = e^{-\frac{1}{2}L_i\theta_i} \tag{2}$$

where L_i is the axis of rotation for the i^{th} joint given by

$$L_1 = L_{1,0} \text{ and } L_i = R_{i-1}...R_1 L_{i,0} R_1...R_{i-1} \quad \text{for } i=2,...,n.$$

Then, the actual orientation of the last link A is obtained of the form

$$A = R_n...R_1 A_0 \tilde{R}_1...\tilde{R}_n \tag{3}$$

where n is the number of joints (degrees of freedom) of the system, A_0 the initial orientation ($\theta_1 = ... = \theta_n = 0$) and $\tilde{R}_i$ is the reverse vector for i^{th} joint given by $\tilde{R}_i = e^{\frac{1}{2}L_i\theta_i}$. Differentiation of equation (2) yields

$$\dot{A} = J_A\dot{\theta} \tag{4}$$

where J_A is the Jacobian matrix. This equation defines the *Differential Kinematics* of the system.

2.2 Discrete State-Space Model

Defining the reference orientation A_d as the vector conformed by the target's point and the origin of the camera frame, a control error variable can be obtained as follows:

$$e_A = A - A_d \tag{5}$$

Then, using (4) and (5), and adding a disturbance term $\gamma(t)$, due to uncertainties in the model, we obtain:

$$\dot{e}_A = J_A\dot{\theta} - \dot{A}_d + \gamma(t)\,. \tag{6}$$

Assuming the terms $\dot{A}_d$ and $\gamma(t)$ are unknown, representing them in a perturbation vector $d(\theta,t)$ and considering $\dot{\theta}$ as the control vector U, the equation (6) can be reformulated as

$$\dot{e}_A = J_A U - d(\theta,t)\,. \tag{7}$$

We assume that function $d(\theta,t)$ is smooth and bounded by known positive scalar function

$$\left\| d(\theta,t) \right\| < \beta_2(\theta,t)\,. \tag{8}$$

Using Euler's discretization, and defining the state-space variables as follows

$$x_{1,k} = \theta_k$$

$$x_{2,k} = A_k$$

we can obtain the discrete state-space model for the system as

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + Tu_k \\ x_{2,k+1} &= x_{2,k} + TJ_{A,k}u_k \end{aligned} \tag{9}$$

where $x_{i,k} = x_i(kT)$ for $i = \{1,2\}$, and T is the sample time. Defining the output of the system as $y_k = x_{2,k}$ and the discrete error variable for the system (9) as

$$e_{A,k} = y_k - A_{d,k} \tag{10}$$

where $A_{d,k}$ is the discrete version of the reference vector, then the dynamics for the error system can be obtained using (9) and (10) as

$$e_{A,k+1} = x_{2,k} + TJ_{A,k}u_k - d_k \tag{11}$$

with d_k as the discretization of perturbation vector.

Now, the problem considered here is to design a Discrete Integral Sliding Mode Controller (DISMC) that ensures visual object tracking in despite of external disturbance d_k.

3 DISM Controller Design

Consider the sliding function s_k as

$$s_k = e_{A,k} + z_k \tag{12}$$

where z_k is the integral variable which is given by the following equation

$$z_{k+1} = z_k - TGe_{A,k} \tag{13}$$

electing $z_0 = -e_0$ to ensure sliding mode occurrence on the sliding manifold $s_k = 0$ from initial instance and G is a design parameter.

From (11) and (12), the projection motion of the system on the subspace s_k can be obtained of the form

$$s_{k+1} = x_{2,k} + TJ_{A,k}u_k - d_k + z_{k+1} \tag{14}$$

We define the control as

$$u_k = -\frac{J_{A,k}{}^{+}}{T}(x_{2,k} - \hat{d}_k + z_k - TGe_k + Ke_k) \tag{15}$$

where $J_{A,k}{}^{+}$ is the pseudo-inverse of $J_{A,k}$, $\hat{d}_k$ is the estimation of perturbation term and K is a constant which will be defined later. From (6) and (7) we know that d_k is the derivative of reference vector A_d, so we can choose its estimation as

$$\hat{d}_k = A_{d,k} - A_{d,k-1} \tag{16}$$

Then, the closed loop system (11),(14) and (15) becomes

$$\begin{aligned} e_{k+1} &= (TG-K)e_k - z_k + \varphi_k \\ s_{k+1} &= \varphi_k - Ke_{A,k} \end{aligned} \tag{17}$$

where $\varphi_k = \hat{d}_k - d_k$, and we can formulate a theorem as follows:

Theorem 1*. If the assumption (8) holds, the control law*

$$u_k = -\frac{J_{A,k}{}^{+}}{T}(x_{2,k} - \hat{d}_k + z_k - TGe_k + Ke_k) \tag{18}$$

is constructed, the inequality

$$\frac{2}{T-2} < G < 0 \tag{19}$$

holds and

$$K = (T-1)G \tag{20}$$

then a solution of the error dynamics (11) converges asymptotically to a vicinity of zero, and this vicinity is bounded by $\varphi_k - \varphi_{k-1}$.

The proof of the Theorem 1 can, unfortunately, not be included due to its length.

Thus, the control objective is fulfilled and the SVS with the proposed discrete controller performs tracking of the target.

4 Application for a Pan-Tilt Unit

The simulations results of this work were obtained applying the designed controller to a Pant-Tilt Unit (PTU). The base frame is defined by the unit vectors $\{e_1, e_2, e_3\}$ .The angles vector is defined as $\theta = [\theta_1, \theta_2]^T$. For simplicity is assumed that the orientation of the second link A is identical to the orientation of the principal axis of right camera, so there is no need of Hand-Eye calibration. The kinematical model for the PTU was obtained as follows. The axes of rotation of the PTU are given by $L_{1,0} = e_3$ and $L_{2,0} = e_1$, the actual orientation A and the Jacobian matrix J_A are calculated as

$$A = \begin{bmatrix} y_1 y_2 \\ -c_1 y_2 \\ c_2 \end{bmatrix} \text{ and } J_A = \begin{bmatrix} y_2 c_1 & y_1 c_2 \\ y_1 y_2 & -c_1 c_2 \\ 0 & -y_2 \end{bmatrix}.$$

where $y_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$, $i = 1, 2$.

In order to obtain a real reference vector $A_{d,k}$, a color segmentation of a moving target in a SVS was developed. The SVS is composed of two Flea® cameras from Point Grey Research Inc. mounted on a metal bar as depicted in Figure 1. The segmentation was based on HSV color space (Hue-Saturation-Value). First, a calibration process for the SVS is realized. Then, the HSV parameters from the target are obtained and compared with the HSV values from the images, resulting in the segmentation of the target in the images.

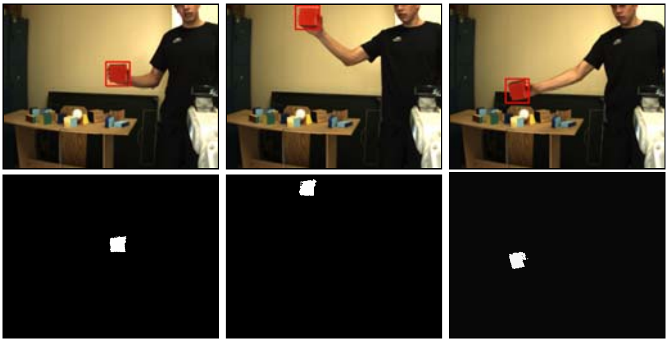


Fig. 2. Color Segmentation (Right Camera)

From segmentation the target is located with 2D coordinates for each image; then the 3D vector V_s, defined from the center of the right camera to the target T_p, is obtained. Figure 3 shows samples of the segmentation process (right column) and the corresponding locations in the image (left column) for the right camera. In total, 45 pairs of images were captured with a sample time of $T = 220\,\text{miliseconds}$. The control parameter for DISMC algorithm is $G = -0.5$ A parallel discrete PID algorithm, defined by the following equation

$$u_k = u_{k-1} - J^{+}_{A,k}[K_p t_s e_k + \frac{K_i}{t_s}(e_k - 2e_{k-1} + e_{k-2}) + K_d(e_k - e_{k-1})]$$

was applied in simulation and the control gains used were

$$K_p = 15,\ K_i = 0.5,\ K_d = 1.$$

The disturbance term γ_k and initial conditions used in both simulations were

$$\gamma_k = \begin{bmatrix} 0.5 \\ 0.2{+}0.3\sin(\pi kT) \\ \text{-}3{+}0.2\cos(kT) \end{bmatrix},\ \theta_0 = \begin{bmatrix} \pi/2 \\ \pi/3 \end{bmatrix},$$

and the sample time 220 milliseconds.

5 Simulation Results

Figure 3 shows the three components of the orientation vector and their references for DISMC and PID algorithms, respectively. It can be appreciated that the goal of control is fulfilled in both controllers, since the objectives are accomplished. However, the convergence in DISCM is smoother than PID, and less oscillatory. The error variables converge to a vicinity of zero in less time (settling time) for DISMC than for

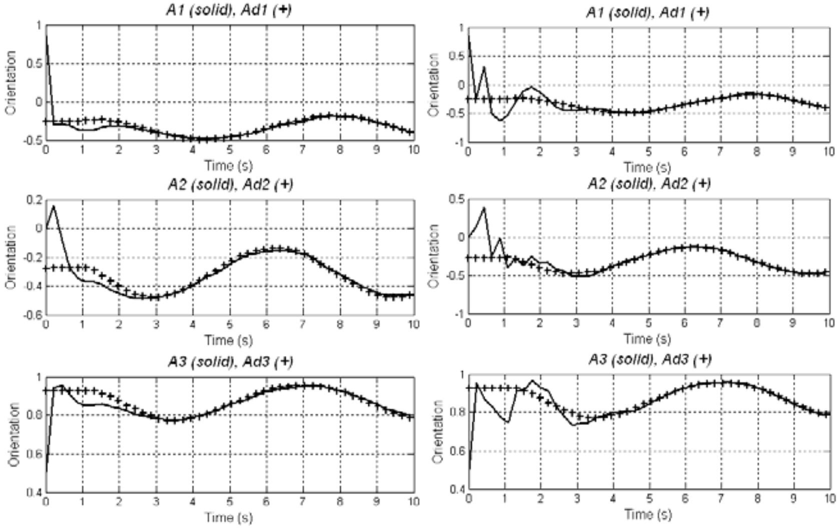


Fig. 3. Orientation Variables and References for DISMC (Left) and for PID controller (Right)

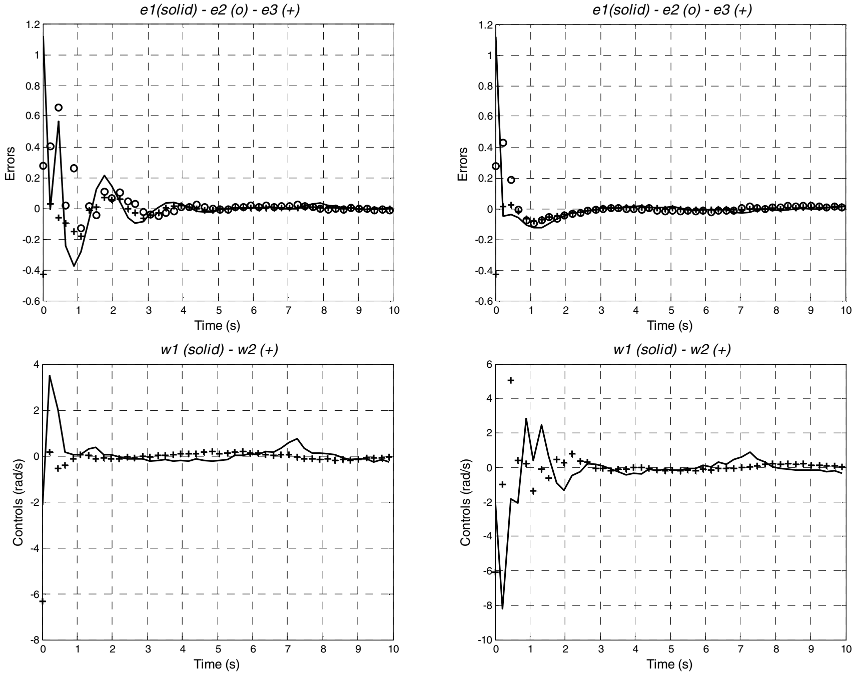


Fig. 4. Error variables in DISMC (Top Left). Error variables in PID controller (Top Right). Control signals in DISMC (Bottom Left). Control signals in PID controller (Bottom Right).

PID, this can be observed in Figure 4. Also, a lower overshoot and a better transient response are noted in DISMC simulation. The angular velocities (control signals) of the joints of the PTU are shown in Figure 4 as well. Note the lower magnitude of the control signals in DISMC and the absence of high frequency components due to the use of a continuous control instead sign function, in comparison with standard Sliding Mode Control.

6 Conclusions

A Discrete Integral Sliding Mode (DISM) controller was designed for Visual Object Tracking and a kinematic model for a PTU was obtained using rotors in Geometric Algebra $G_{3,0,0}$, which is a simpler method than using matrices. The proposed algorithm demonstrates a satisfactory performance in output tracking problem, since it achieves a reduced steady state tracking error. A comparison with discrete PID controller was made and DISMC showed a better performance in aspects like softness in control signals, lower magnitude in transient response, settling time and overshoot. The procedure for obtain the DISM controller is simple, and it can be applied to any kind of serial kinematical structure, moreover, the use of a continuous control law allows to ensure chattering-free SM motion. So, it can be concluded that ISMC is a good approach to solve the Visual Object Tracking problem.

References

1. Kragic, D., Miller, A.T., Allen, P.K.: Real-time tracking meets online grasp planning. In: International Conference on Robotics and Automation (ICRA), Seoul, Republic of Korea, pp. 2460–2465 (2001)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision (ICCV), Corfu, Greece, pp. 1150–1517 (1999)
3. Swain, M.J., Ballard, D.H.: Color Indexing. International Journal of Computer Vision 7, 11–32 (1991)
4. Craig, J.J.: Adaptive Control of Mechanical Manipulators. Ed.Addison-Wesley, Reading (1988)
5. Ozaki, T., Susuki, T., Furuhashi, T., Okuma, S., Uchikawa, Y.: Trajectory Control of robotic Manipulator Using neural Networks. IEEE Transactions on Industrial Electronics 39(6), 555–570 (1992)
6. Utkin, V.I., Guldner, J., Shi, J.: Sliding Mode Control in Electromechanical Systems. Ed. Taylor and Francis, UK (1999)
7. Utkin, V.I., Shi, J.: Integral sliding mode in systems operating under uncertainty. In: IEEE Conference on Decision and Control CDC 1996, Kobe, Japan (1996)
8. Angeles, J.: Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms, 2nd edn. Ed. Springer, USA (2002)
9. Utkin, V.: Sliding mode control in discrete-time and difference systems. In: Zinober, A.S.I. (ed.) Variable Structure and Lyapunov Control, ch. 5, vol. 193, pp. 87–107. Springer, New York (1994)
10. Zamora, J., Bayro, E.: Kinematics and Differential Kinematics of Binocular Heads. In: Proc. of the Int. Conf. of Robotics and Automation ICRA 2006, Orlando Florida, USA, pp. 4130–4135 (2006)

Machine Learning and Geometric Technique for SLAM

Miguel Bernal-Marin and Eduardo Bayro-Corrochano

Department of Electrical Engineering and Computer Sciences,
CINVESTAV Unidad Guadalajara, Av. Cientfica 1145, Col. El Bajo,
Zapopan, Jalisco 45015, Mexico
{mbernal,edb}@gdl.cinvestav.mx

Abstract. This paper describes a new approach for building 3D geometric maps using a laser rangefinder, a stereo camera system and a mathematical system the Conformal Geometric Algebra. The use of a known visual landmarks in the map helps to carry out a good localization of the robot. A machine learning technique is used for recognition of objects in the environment. These landmarks are found using the Viola and Jones algorithm and are represented with their position in the 3D virtual map.

1 Introduction

Mobile robots are equipped with multiple input devices to sense the surrounding environment. The laser rangefinder is widely used for this task due to its precision, and its wide capture range. In this paper we merged the data obtained by the laser and the stereo camera system to build a 3D virtual map with the shapes obtained by these devices. The 3D objects seen by the stereo camera system can be modeled by geometric entities, which are easy to represent and to combine. Some of these 3D objects can act as a landmarks for the robot navigation and relocalization. Line segments are used to build a 3D map and they are the most widely used features [1] [2].

Using the Conformal Geometric Algebra we can represent different geometric shapes including the line segments (as a pair of points) and the data captured by the stereo camera system (landmarks as labeled spheres). This framework also allows us to formulate transformations (rotation, translation) using spinors or versors.

2 Geometric Algebra

The Geometric algebra $\mathcal{G}_{p,q,r}$ is constructed over the vector space $\mathcal{V}^{p,q,r}$, where p,q,r denote the signature of the algebra; if $p \neq 0$ and $p = r = 0$, the metric is Euclidean; if only $r = 0$, the metric is pseudo Euclidean; if $p \neq 0$, $q \neq 0$, $r \neq 0$, the metric is degenerate. The dimension of $\mathcal{G}_{n=p+q+r}$ is 2^n, and $\mathcal{G}_n$ is

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 851–858, 2009.

constructed by the applications of the *geometric product* over the vector basis $\mathsf{e_i}$. The geometric product between two vectors **a**,**b** is defined as

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$$

and the two parts; the inner product $\mathbf{a} \cdot \mathbf{b}$ is symmetric part, while the wedge product (outer product) $\mathbf{a} \wedge \mathbf{b}$ is the antisymmetric part.

In $\mathcal{G}_{p,q,r}$ the geometric product of two basis is defined as

$$\mathsf{e_i e_j} := \begin{cases} 1 \in \mathbb{R} & \text{for } i = j \in \{1, \ldots, p\} \\ -1 \in \mathbb{R} & \text{for } i = j \in \{p+1, \ldots, p+q\} \\ 0 \in \mathbb{R} & \text{for } i = j \in \{p+q+1, \ldots, n\} \\ \mathsf{e_{ij}} = \mathsf{e_i} \wedge \mathsf{e_j} & \text{for } i \neq j. \end{cases}$$

this lead in a basis for $\mathcal{G}_n$ that contains elements of different grade called *blades* (e.g. scalars, vectors, bivectors, trivectors, etc.): $1, \{\mathsf{e_i}\}, \{\mathsf{e_i} \wedge \mathsf{e_j}\}, \{\mathsf{e_i} \wedge \mathsf{e_j} \wedge \mathsf{e_k}\}, \cdots, \mathsf{e_1} \wedge \mathsf{e_2} \wedge \cdots \wedge \mathsf{e_n}$ which is called *basis blade*; where the elements of maximum grade is the pseudoscalar $\mathtt{I} = \mathsf{e_1} \wedge \mathsf{e_2} \wedge \ldots \wedge \mathsf{e_n}$. A linear combination of basis blades, all of the same grade k, is called k-vector. The linear combination of such k-vectors is called *multivector*, and multivectors witch certain characteristics represent different geometric objects or entities (as points, lines, planes, circles, spheres, etc.), depending on the GA where we are working (for example, a point (a, b, c) is represented in $\mathcal{G}_{3,0,0}$ [the GA of the 3D-Euclidean space $\mathcal{E}^3$] as $\mathtt{x} = a\mathsf{e_1} + b\mathsf{e_2} + c\mathsf{e_3}$, however a circle can not be defined in $\mathcal{G}_{3,0,0}$, but it is possible to define it in $\mathcal{G}_{4,1,0}$ (CGA) as a 4-vector $\underline{z} = \underline{s_1} \wedge \underline{s_2}$ [the intersection of two spheres in the same space]). Given a multivector M, if we are interested in extracting only the blades of a given grade, we write $< M >_r$ where r is the grade of the blades we want to extract (obtaining an homogeneous multivector *M'* or a r-vector).

The *dual* $\mathbf{X}^*$ of a r-blade $\mathbf{X}$ is defined by $\mathbf{X}^* = \mathbf{X}\mathtt{I}_\mathsf{n}^{-1}$. It follow that the dual of a r-blade is an $(n-r)$-blade.

The *reverse* of any multivector M is defined as

$$\langle \widetilde{M} \rangle_i = (-1)^{\frac{i(i-1)}{2}} \langle M \rangle_i, \text{ for } M \in \mathcal{G}_n, 0 \leq i \leq n. \tag{1}$$

The reader should consult [3] to detailed explanation about CGA and its applications.

2.1 Conformal Geometric Algebra

To work in Conformal Geometric Algebra (CGA) $\mathcal{G}_{4,1,0}$ means to embed the Euclidean space in a higher dimensional space with two extra basis vectors which have particular meaning; in this way we represent particular entities of the Euclidean space with subspaces of the conformal space. The extra vectors we add are e_+ and e_-, defined by the properties $\mathsf{e}_+{}^2 = 1, \mathsf{e}_-{}^2 = -1, \mathsf{e}_+ \cdot \mathsf{e}_- = 0$. With this two vectors, we define the null vectors $\mathsf{e_0} = \frac{1}{2}(\mathsf{e}_- - \mathsf{e}_+)$ and $\mathsf{e} = \mathsf{e}_- + \mathsf{e}_+$ interpreted as the origin and the point at infinity, respectively. From now on and in the rest of the paper, points in the 3D-Euclidean space are represented in lowercase, while conformal points in underline letters; also the conformal entities will be expressed in the *Outer Product Null Space* (OPNS) (noted with an asterisk beside, also know as the dual of the entity), and no in the *Inner Product Null Space* (IPNS) (without asterisk) unless it is specified explicitly. To go from OPNS to IPNS we need to multiply the entity by the pseudoscalar.To map a point $\mathbf{x} \in \mathcal{V}^3$ to the Conformal space in $\mathcal{G}_{4,1}$ (using IPNS) we use

$$\underline{x} = \mathbf{x} + \frac{1}{2}\mathbf{x}^2\mathsf{e} + \mathsf{e}_0 \tag{2}$$

Applying the wedge operator "$\wedge$" on points, we can express new entities in CGA. All geometric entities from CGA are show in the table 1 for a quick reference.

The pseudoscalar in CGA $\mathcal{G}_{4,1,0}$ is defined as $\mathtt{I} = \mathtt{I}_\mathtt{E}\mathtt{E}$, where $\mathtt{I}_\mathtt{E} = \mathsf{e}_1\mathsf{e}_2\mathsf{e}_3$ is the pseudoscalar from $\mathcal{G}_3$ and $\mathtt{E} = \mathsf{e}_+\mathsf{e}_-$ is the pseudoscalar from the Minkowski plane.

In GA there exist specific operators to model rotations and translations called *rotors* and *translators* respectively. In CGA such operator are called *versor* and are defined by (3) being $\mathbf{R}$ the rotor, $\mathbf{T}$ the translator.

$$\mathbf{R} = e^{-\frac{1}{2}\underline{l}\theta}; \;\; \mathbf{T} = e^{\frac{\mathsf{e}\mathbf{t}}{2}}, \tag{3}$$

where the *rotation axis* $\underline{l} = l_1\mathsf{e}_{23} + l_2\mathsf{e}_{31} + l_3\mathsf{e}_{12}$ is a unit bivector which represents a line (in IPNS) through the origin in CGA, θ is the rotation angle, $\mathbf{t} = t_1\mathsf{e}_1 + t_2\mathsf{e}_2 + t_3\mathsf{e}_3$ is the translation vector in $\mathcal{V}^3$. The equations (3) can also be expressed as

$$\mathbf{R} = cos\left(\frac{\theta}{2}\right) - sen\left(\frac{\theta}{2}\right)\underline{l}; \;\; \mathbf{T} = (1 + \frac{\mathsf{e}\mathbf{t}}{2}) \tag{4}$$

Table 1. Entities in CGA

Entity	IPNS	OPNS
Sphere	$\underline{s} = \mathbf{p} + \frac{1}{2}(\mathbf{p}^2 - \rho^2)\mathsf{e} + \mathsf{e}_0$	$\underline{s}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \underline{d}$
Point	$\underline{x} = \mathbf{x} + \frac{1}{2}\mathbf{x}^2\mathsf{e} + \mathsf{e}_0$	$\underline{x}^* = (-\mathtt{E}\mathbf{x} - \frac{1}{2}\mathbf{x}^2\mathsf{e} + \mathsf{e}_0)\mathtt{I}_\mathtt{E}$
Plane	$\underline{P} = \mathbf{N}\mathtt{I}_\mathtt{E} - d\mathsf{e}$ $\mathbf{N} = (\mathbf{a} - \mathbf{b}) \wedge (\mathbf{a} - \mathbf{c})$ $d = (\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c})\mathtt{I}_\mathtt{E}$	$\underline{P}^* = \mathsf{e} \wedge \underline{a} \wedge \underline{b} \wedge \underline{c}$
Line	$\underline{L} = \underline{P_1} \wedge \underline{P_2}$ $= \mathbf{r}\mathtt{I}_\mathtt{E} + \mathsf{e}\mathbf{M}\mathtt{I}_\mathtt{E}$ $\mathbf{r} = \mathbf{a} - \mathbf{b}$ $\mathbf{M} = \mathbf{a} \wedge \mathbf{b}$	$\underline{L}^* = \mathsf{e} \wedge \underline{a} \wedge \underline{b}$
Circle	$\underline{z} = \underline{s_1} \wedge \underline{s_2}$ $\underline{s_z} = (\mathsf{e} \cdot \underline{z})\underline{z}$ $\rho_{\underline{z}} = \frac{\underline{z}^2}{(\mathsf{e} \wedge \underline{z})^2}$	$\underline{z}^* = \underline{a} \wedge \underline{b} \wedge \underline{c}$
P-pair	$\underline{PP} = \underline{s_1} \wedge \underline{s_2} \wedge \underline{s_3}$	$\underline{PP}^* = \underline{a} \wedge \underline{b}$

due to the exponential properties. Such operator are applied to any entity of any dimension by multiplying the entity by the operator from the left, and by the *reverse* of the operator from the right, as show in (5).

$$\underline{x'} = \sigma \underline{x} \widetilde{\sigma} \tag{5}$$

where $\underline{x}$ is any entities mentioned in table 1, and σ is a versor (*rotor*, *translator* or *motor* mentioned below). Using (5) is easily to transform any entities from CGA (points, point-pair, lines, circles, planes, spheres), not only points as is usual in other algebras.

In CGA it is possible to use the rotors and translator to express general rotation and screw motions in space. To model a screw motion, the entity has to be translated during a general rotation with respect to the rotation axis. The implementation consecutive of a translator and rotor can be written as the product of them. Such operator is called *motor* and expressed as

$$\mathbf{M} = \mathbf{TR} \tag{6}$$

The translator, rotor and motor (all of them *versors*) are elements from $\mathcal{G}^+_{4,1}$, and they defines an algebra called *motor algebra*. This algebra greatly simplifies the successive computation of rotations and translation, applying only the geometric product in consecutive versors, giving the final result another versor of this algebra, where all the transformations are together in one element.

Vector calculus is a coordinate dependent mathematical system and its cross product can not be extended to higher dimensions. The representation of geometric primitives is based in lengthy equations and for linear transformations one uses matrix representation with redundant coefficients. In contrast conformal geometric algebra a coordinate free system provides a fruitful description language to represent primitives and constraints in any dimension and by using successive reflections with bivectors one builds versors to carry out linear transformations avoiding redundant coefficients.

3 3D Map Building

Using an equipped mobile robot with a laser rangefinder sensor and stereo camera system mounted on a pan-tilt head, each one with their own coordinate system. We apply *the method of hand-eye calibration* [4] to get the center coordinates of each devices related to a global robot coordinate system. Using the perpendicular line to plane (x, y) as rotation axis and the angle of rotation of the robot, and adding a third fixed coordinate to the robot's movement in the plane we can apply this values in (3) to make $\mathbf{T_{pos}}$ and $\mathbf{R_{pos}}$ that represent the movement of the robot in the 3D environment.

Line segments from range points are extracted using recursive line splitting method as show in [1] [2], this is a speedy and correctness algorithm that performs *divide-and-conquer* algorithms [5]. For every endpoints of the line segments, we maps them to CGA to get the pair of points entity (see table 1) and store in a map. To translate the position of the entities we use the motor ([6]) which is defined as the translation and rotation of the mobile robot.

To get the position of any entity extracted by the stereo camera system or laser rangefinder in the 3D environment, we apply a specific transformation using motors in CGA. Then place the entity in the 3D map.

4 Getting 3D Positions Based on Visual Landmarks

A landmark literally is a geographic feature used by explorers and others to find their way back or move through an area. In the map building process a mobile robot can use these landmarks to remember the place where it was before while it explore its environment. Also the landmarks can be used to find robot position in a previous building map facilitating the relocalization. As we are using a camera stereo system, the 3D position of any object can be also calculated and it can be represented in the 3D virtual environment. Using these objects as a landmarks, the robot gets its relative position.

4.1 Machine Learning Phase

A natural or artificial landmark located in the actual environment helps to the mobile robot to know its position on the map. Viola and Jones present a new and radically faster approach to face detection based on the AdaBoost algorithm from machine learning [6], and this approach can be used to detect our statics landmarks. Once the landmarks have been selected and trained, the mobile robot can use them to navigate in the environment performs the Viola an Jones algorithm. If a landmark is found we get a sub-image I_L from the left camera image. This I_L is the region of the image where the landmark was found (fig. 2).

When a landmark is identified in one image (left camera), we must be sure that the landmark is in the other image as well (right camera of the stereo camera system). To get the 3D position, the landmark must be detected in both images. The landmark in the right image is also detected by Viola and Jones algorithm, and identify its region by a sub-image I_R (fig. 1).

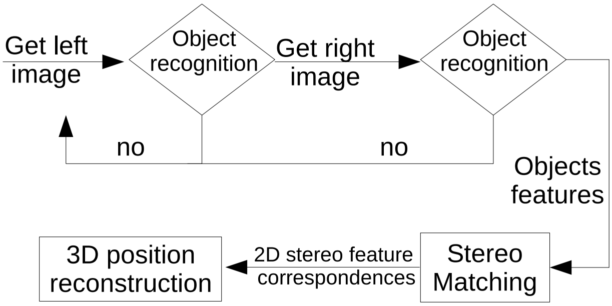


Fig. 1. The flowchart of the landmark position estimation

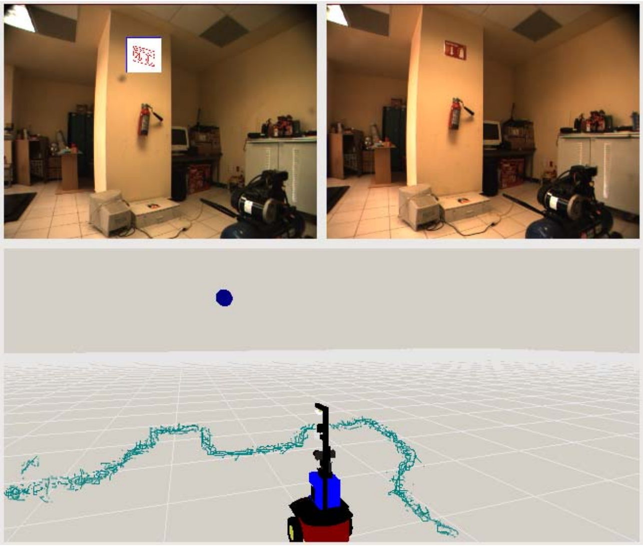


Fig. 2. Mobile robot founding landmarks while it is navigating its environment. On the top see stereo view (on left image a sign landmark found in white rectangle) and its representation in the 3D map.

4.2 Landmark Position Estimation

When we talk about the landmark position estimation, we are looking for the 3D location of these landmark in the environment and not for the pose (position and orientation) of the object found. Figure 1 illustrates the flowchart of the landmark position estimation.

Getting the landmark identified in both images, we proceed to calculate the points of interest. To do this we use Canny edge detection operator on I_L and a correlation. A number of correlation-based algorithms attempt to find points of interest on which to perform the correlation. In fact, the normalization embodied into the *Normalized Cross Correlation* (NCC) and *Zero Mean Normalized Cross Correlation* (ZNCC) allows for tolerating linear brightness variations. Further more, thanks to the subtraction of the local mean, the ZNCC provides better robustness than the NCC [7] since it tolerates uniform brightness variations as well.

Correspondences of an image patch are searched for along the epipolar line by calculating the ZNCC only in a given interval $(d_{min}, \ldots, d_{max})$ of so-called disparities [8] [9]. The term *disparity* denotes the Euclidean distance from one point on the epipolar line to a given point in the other camera image [10]. A small disparity represents a large distance to the camera, a large value a small distance (parallax).

When all the points are matched in both images we proceed to calculate its 3D position using the triangulation. Then we integrate this set of points to get its center of gravity and place the center of a sphere on it. The radius of the sphere is calculated taking the highest number of points of the landmark. The sphere is stored in the 3D virtual map using CGA and it is labeled as a landmark.

5 Experiments

The fig. 3 shows some signs as landmarks in a laboratory. A sign was trained with Viola-Jones algorithm. The result is: where a sing is placed in the wall, it is recognised by the robot, and place it at its 3D position in the environment (fig. 4). The first and last image of figure 3 represent the same landmark, but taken on different place of the laboratory.

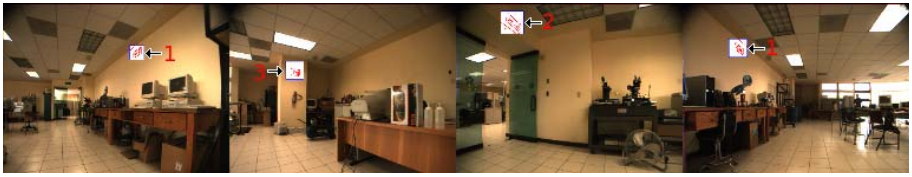


Fig. 3. Signs as Landmarks in a laboratory

The figure 4 shows the map built with lines, the raw data of the laser (points), and the landmarks found.

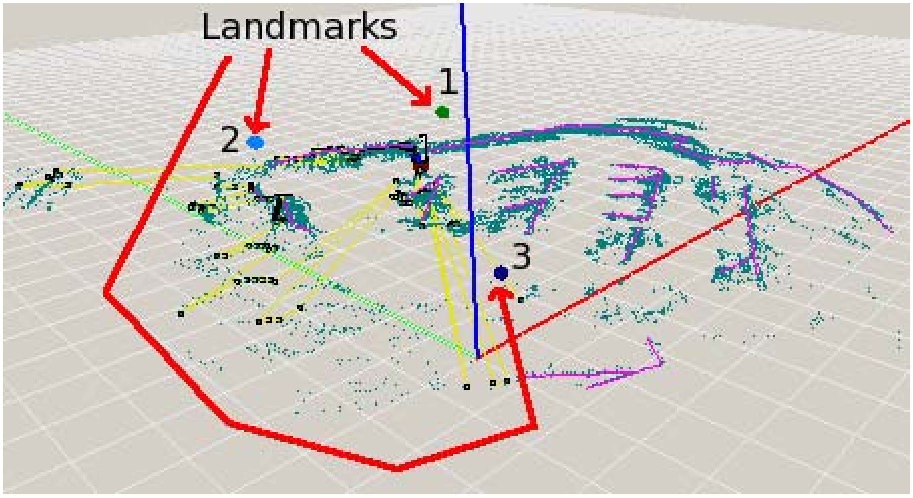


Fig. 4. 3D representation of the landmarks and the laser readings as lines and points cloud

6 Conclusions

In this paper the authors have shown the use of geometric entities in Conformal Geometric Algebra (CGA) for modeling input data for a 3D virtual environment, in this way merging in a global coordinate system, the laser rangefinder and stereo camera system (mounted over a pan-tilt unit). The machine learning technique is used for the object's recognition. The detected objects are used as a landmarks witch greatly help in the interaction with the environment. The experiments with a real robot validate our method. We believe that our approach can be of great use for mobile robots or upper body humanoids installed on mobile platforms.

References

1. Zhang, L., Ghosh, B.K.: Line segment based map building and localization using 2d laser rangefinder. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2538–2543 (2000)
2. Siadat, A., Kaske, A., Klausmann, S., Dufaut, M., Husson, R.: An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation. In: Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications, pp. 153–158 (1997)
3. Bayro-Corrochano, E.: Robot perception and action using conformal geometry. In: Bayro-Corrochano, E. (ed.) The Handbook of Geometric Computing. Applications in Pattern Recognition, Computer Vision, Neurocomputing and Robotics, ch. 13, pp. 405–458. Springer, Heidelberg (2005)
4. Bayro-Corrochano, E., Daniilidis, K., Sommer, G.: Motor algebra for 3d kinematics: The case of the hand-eye calibration. Journal of Mathematical Imaging and Vision archive 13, 79–100 (2000)
5. Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. Auton. Robots 23(2), 97–111 (2007)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December 2001, pp. 511–518 (2001)
7. Di Stefano, L., Mattoccia, S., Tombari, F.: ZNCC-based template matching using bounded partial correlation. Pattern Recogn. Lett. 26(14) (2005)
8. Faugeras, O., et al.: Real-time correlation-based stereo: algorithm, implementation and applications. INRIA Technical Report no. 2013 (1993)
9. Azad, P., Gockel, T., Dillmann, R.: Computer Vision: Principles and Practice. Ed. Elektor Electronics (2008)
10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Ed. Cambridge University Press, Cambridge (2004)

Compression and Key Feature Extraction for Video Transmission

Esteban Tobias Bayro Kaiser[1], Eduardo Correa-Arameda[1],
and Eduardo Bayro-Corrochano[2]

[1] University of Tarapacá, Escuela universitaria de ingeniería eléctrica y electrónica
Arica-Chile
locotoebk@hotmail.com, ecorrea@ut.edu.ch
[2] CINVESTAV, Unidad Guadalajara, Departamento de Ingeniería Eléctrica y
Ciencias de la Computación, Jalisco, México
edb@gdl.cinvestav.mx

Abstract. This paper presents a gray scale image compression method using the Wavelet Transform and key feature detection for mobile phone video transmission. The major contribution of this work is to show the application of the wavelet transform in image compression and to add a new method to reduce redundant information in video transmission which is key feature detection. An algorithm is designed in Matlab to accomplish this task using a face to face video.

Keywords: Image processing, Compression, Wavelet transform, Key feature extraction, Face to face video.

1 Introduction

In the transmission of multimedia whether it is data, video or images, bandwidth and storage capacity are very important factors. Regardless of the constant improvement of storage devices and increasing the bandwidth, compression of information is still very necessary. There are several techniques for compressing information. These differ mainly in the amount of information that can be compressed and the amount of loss that is obtained when trying to restore them.

In this paper we will study the transmission of video between two people, this may be between two computers or between two cell phones; this type of transmission is known as face to face. The video that is used in this work consists of gray-scale images whose main feature is the face of the person. The images must be compressed or reduced in size to reduce the required bandwidth in the transmission [1].

For image compression there are several techniques available. In this paper the compression of the images is accomplished with the wavelet Transform.

In addition to analyse the image compression with the wavelet transform a method will be studied which transmits only the image key features [2]. These features are transmitted in sub-images. In a face to face transmission (for example: between cell phones), the key features are the most notable, and these are:

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 859–866, 2009.

eyes and lips movements, therefore a technique is developed to locate and extract these features.

2 Compression of Images with the Wavelet Transform

Data compression refers to the process of reducing the volume of data needed to represent a certain amount of information, in this case gray level images. The compression is accomplished with the wavelet transform and Huffman coding. Huffman coding is an entropy encoding algorithm used for lossless data compression. The term coding refers to the use of a variable-length code table for encoding a source symbol where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

3 Key Features Extraction

The key features extraction of an image can be useful for reducing the amount of information in a video transmission. This information comes within a matrix, so if the size can be reduced, then the image has a representation with less information.

In a video each image compared with its previous suffers very little change, this feature can be harnessed to just focus on the analysis and transmission of these. When the video comes from a conversation between two people focusing on the head with a single fund the major changes affecting the image are the eye and lip movements.

If the eyes and lips extraction is success, then the full image can be represented by a previous image and the extracted sub-image.

For a face to face video, the steps for key features extraction are:

- Eye position detection, Image alignment, Sub-images extraction and Difference estimation between images.

3.1 Eye Position Detection

For eye position detection there are several steps that are listed below.

Key features Position
In order to find key features in the image, we binarize the gray scale image by applying a Sobel filter and a thresholding to the edge image [3].

In Fig. 3.1 b) it can be observed that eyes and mouth are dominating features in the image, thus there can be found a method to obtain the position of these key features.

These positions can be found through horizontal and vertical projections, analyzing the maximum and minimum of these functions. Then it is possible to determine positions of certain features such as eyes, mouth, nose, upper and lower head and side edges. Assuming $I(x, y)$ be the intensity of a image, the

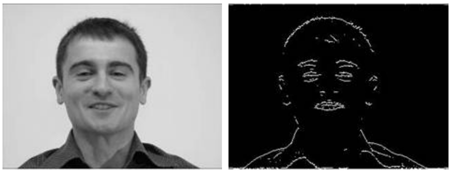


Fig. 3.1. a) Gray scale image, **b)** Edge detection with a Sobel filter and a T =0.5804 thresholding

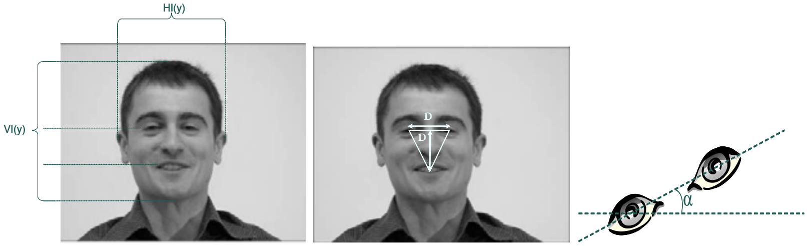


Fig. 3.2. a) Points where the horizontal and vertical projected are analyzed. b) Geometric relationship between eyes and mouth. c) Rotation angle.

equations are: $HI(x) = \sum_{x=1}^{m} I(x,y)$, $VI(y) = \sum_{y=1}^{n} I(x,y)$, where the HI corresponds to the horizontal and VI vertical projection (see Fig. 3.2 a)).

Geometric relationship between eyes and mouth
To obtain the position of the mouth, the geometric relationship between eyes and mouth can be used. Knowing that the distance between the eyes is equal to the distance from eye to mouth, it is possible to find the respective coordinates with an isosceles triangle on the face (see Fig. 3.2 b)).

3.2 Image Alignment and Rotation Angle

The alignment can be accomplished by rotation. To determine the rotation angle it is necessary to consider the following: If the goal is always to have the eyes lined up in a horizontal line then it necessary to find the angle between the horizontal and eyes, as shown in Fig. 3.2 c).

To find this angle requires the eyes coordinates, left eye represented by the point (x_{left}, y_{left}) and the right eye by (x_{right}, y_{right}). The angle can be determined by $\alpha = \tan^{-1}(\frac{y_{right}-y_{left}}{x_{right}-y_{left}})$. It is worth mentioning that the image can be rotated both directions.

3.3 Sub-image Extraction and Difference Estimation between Images

Once the image has been aligned, the image can be divided into sub-matrices; as it used the coordinates of the eyes and mouth, the adequate estimated size matrices will contain the entire feature. The centers of these sub-matrices are the coordinates for each feature.

To estimate the difference between each image or sub-image there must be a cost function, and the most appropriate is MAE (Mean Absolute Error). $MAE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|I(i,j) - I'(i,j)|$, where I is an image and $I^{'}$ is its previous one. The value delivered by MAE will be positive and real. To decide whether an image changes significantly, it is necessary to calculate the MAE and then find a value that can also be called thresholding, which will be called thresholding of difference T. This value will be the decision point if the sub-image is transmitted or not.

It's important to note that this value of T difference is calculated by considering the visual effect of change that each image has with the previous one. Then the following comparison must be done, if MAE is greater than the difference of T, the image is sent, otherwise there is no need to send it because the difference is very small and it is sufficient to represent the sub-image with its predecessor.

4 Code Results in Matlab

This section shows the analysis results of the algorithms written with Matlab software [4]. The algorithm can be divided into two parts: one considers its earlier phase of pre-transmission, and post-transmission. Both are shown in Figs. 4.1 a) and b).

4.1 Pre-transmission Algorithm

The pre-transmission phase will be put under performance test, to do this, the phase is divided into two parts that allow an individual analysis for each stage. Different types of face images are used, with different characteristics and situations, which helps to determine the functionality and efficiency of the algorithm.

4.2.1 Algorithm for the Extraction of Important Features and Decision Making

The Fig. 4.1 c) shows the algorithm to extract important features and decision-making. In this stage, the image is aligned and focused, and the algorithm is responsible for selecting those parts of the image that are relevant, such as detecting the movements of eyes and lips.

In this case, it is considered image patterns in gray level and taken in front of a single person. The background of these pictures should be simple and continuous, that is, without a background objects that change shape. In addition the person must be looking at the camera at all times.

For a better appreciation each block will be analyzed separately.

Image Analysis

The image size is 135 x 99 pixels.

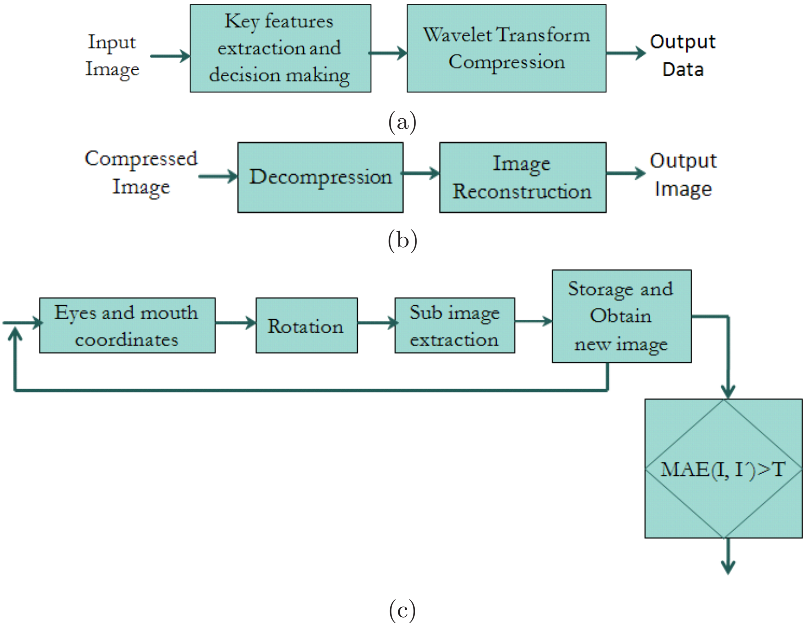


Fig. 4.1. Results of the 3D reconstruction process. a) Pre-transmission algorithm. b) Post-transmission algorithm. c) Algorithm to extract important features and decision-making. Figures a) and b) show that every part of algorithm consists of two major sections, this is done in order to analyze the efficiency of the algorithm.

i) Edge detection. Figure 4.2 shows that the edges of the image are well defined, which indicates that there is no objection to proceed to the next step: the horizontal and vertical projections.

ii) Horizontal and vertical projections. In the horizontal projection of the Fig. 4.3 a) the first 2 maximum are obtained, these indicate the sides of the face. In the vertical projection of Fig. 4.3 b) the maximum value indicates the central line of the eye.

- Maximum horizontal projection: 26 and 77. Maximum vertical projection: 42.

With these points it is possible to obtain two sub-images where each one contains an eye. Then the same procedure is repeated.

iii) Sub-image Edge detection. Note: The left and right refer to as seen by the observer.

In Figs. 4.4 a) and b) the eye is identified easily without other information that could interfere with the projections, with this step any information that is not of interest was removed to find the exact iris coordinates.

iv) Sub-image horizontal and vertical projections.

- Maximum horizontal projection from the left eye: 17 (see Fig. 4.5). Maximum vertical projection from the left eye: 6.

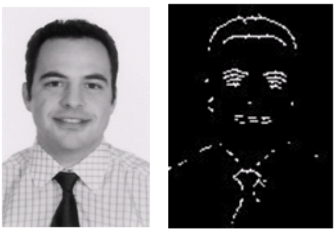


Fig. 4.2. Edge detection with thresholding T = 0.5608

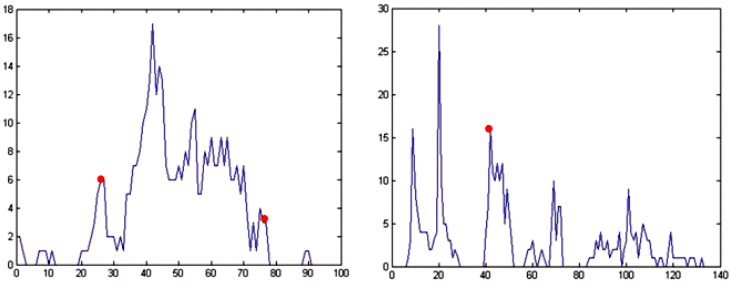


Fig. 4.3. a) Horizontal projection, b) Vertical projection

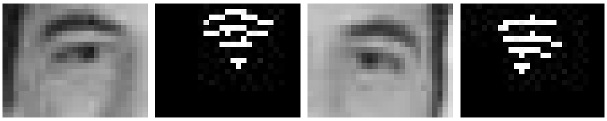


Fig. 4.4. Sub-image edge detection. a) left eye, with thresholding T = 0.3980. b) right eye, with thresholding T = 0.4784.

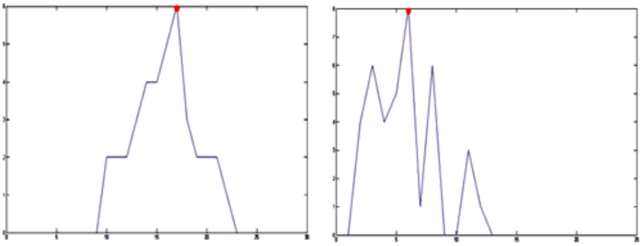


Fig. 4.5. a) Horizontal projection left eye and b) Vertical projection left eye

- Maximum horizontal projection from the right eye: 4 (see Fig. 4.6). Maximum vertical projection from the right eye: 12.

After finding these points, the offset that was created to generate the sub-image is added to obtain the iris location.

- Left eye position: (45, 43). Right eye position: (46, 64).

With these points and the geometric relationship between eyes and mouth, the mouth position can be calculated. Various tests determined that the distance from the center of the eye to the mouth is 1.2 D.

- Mouth position: (70, 52) (see Fig. 4.7).

The same procedure was carried out to various images and the algorithm was right with the coordinates of the features, as shown in Fig. 4.8.

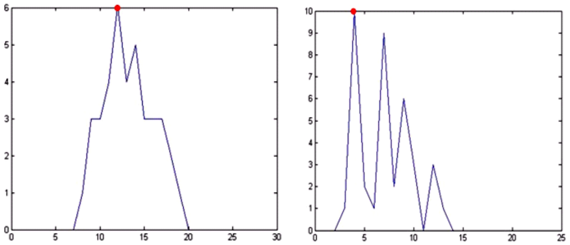


Fig. 4.6. a) Horizontal projection right eye. b) Vertical projection right eye.

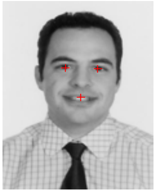


Fig. 4.7. Eyes and mouth position

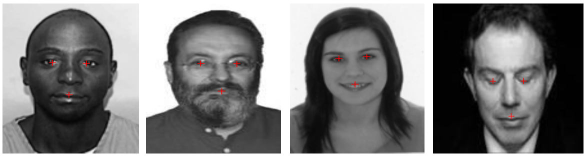


Fig. 4.8. Eyes and mouth positions

5 Conclusions

One of the key factors that is analyzed in this work is to perform the transmission of video images, considering the least amount of information possible from each image, without significant loss of information, in an ideal channel. To achieve the image compression from a video using the wavelet transform an algorithm in Matlab was design resembling a JPEG200 compression [5]. The algorithm is able to compress and decompress images with the least information loss.

The images that were used for this algorithm allowed only the first level of compression, achieving a ratio of no more than 10:1 compression. The image compression that was achieved by extracting sub-images was about 90:1, this result shows that in the image compression key features extraction must be considered.

Acknowledgments. Esteban T. Bayro Kaiser, was supported by Msc Eduardo Correa Arameda and Prof. Eduardo Bayro Corrochano.

References

1. Ravyse, I., Sahli, H.: Facial analysis and synthesis. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 810–820. Springer, Heidelberg (2006)
2. Nixon, M., Aguado, A.: Feature Extraction and Image Processing, 1st edn. Elsevier Ltd., Amsterdam (2002)
3. Gonzalez, R., Woods, R.: Digital Image Processing using Matlab, 1st edn. Pearson Prentice hall, London (2004)
4. Mathworks: http://www.mathworks.com/
5. Usevitch, B.E.: A tutorial on modern lossy wavelet image compression: Foundations of jpeg 2000. IEEE Signal Processing Magazine 18, 22–35 (2001)

XV Robot Vision

Robot Command Interface Using an Audio-Visual Speech Recognition System

Alexánder Ceballos[1,2], Juan Gómez[2,4], Flavio Prieto[3], and Tanneguy Redarce[4,⋆]

[1] Instituto Tecnológico Metropolitano, Medellín, Colombia
[2] DIEEC, Universidad Nacional de Colombia Sede Manizales, Manizales, Colombia
[3] DIMM, Universidad Nacional de Colombia Sede Bogotá, Bogotá, Colombia
[4] Institut National des Sciences Appliquées de Lyon, Lyon, France
alexanderceballos@itm.edu.co, faprietoo@unal.edu.co,
{juan-bernardo.gomez-mendoza,tanneguy.redarce}@insa-lyon.fr

Abstract. In recent years audio-visual speech recognition has emerged as an active field of research thanks to advances in pattern recognition, signal processing and machine vision. Its ultimate goal is to allow human-computer communication using voice, taking into account the visual information contained in the audio-visual speech signal. This document presents a command's automatic recognition system using audio-visual information. The system is expected to control the laparoscopic robot da Vinci. The audio signal is treated using the Mel Frequency Cepstral Coefficients parametrization method. Besides, features based on the points that define the mouth's outer contour according to the MPEG-4 standard are used in order to extract the visual speech information.

Keywords: Speech recognition, MPEG-4, manipulator.

1 Introduction

The da Vinci system is a laparoscopic surgery system which consists of a control console, a stretcher, four robotical arms and a high performance vision system. The control console can be located at the side of the surgery table or even at an adjacent room, enabling the surgeon to use the system without carrying a face mask. While the surgeon observes 3D images through a stereo vision system, both camera and instruments are controlled by joysticks, and the surgeon switches between them with pedals. When driving the camera, the surgeon loses the instruments control, and sometimes it is necessary to reposition them. In order to avoid this situation, the development of an alternative interface for commanding camera movements is desired.

There are several approaches for commanding an endoscope holder robot proposed in literature. Some of them assist the surgeon in endoscope location by

⋆ This work was partially supported by the ECOS Franco-Colombian program (ECOS-Nord/COLCIENCIAS/ICFES/ICETEX), the Bonpland scholarship program, and the CNRS.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 869–876, 2009.

using joysticks [1], [2], pedals [3], voice commands [3], [4], etc.. Others use visual or force feedback and geometrical constraints in order to track tool's location during the intervention [5].

Automatic speech recognition systems (ASR) is an active research field, mainly because noise in the audio signal propose an unresolved challenge to the recognition systems. Carelessness of the speaker, variation in the frequency and duration of the words, grammar subjects, are other factors that also impose some difficulties when performing the voice command recognition [6], [7], [8].

ASRs proposed in [4] and [9] have a high recognition rate and showed that using voice commands is a admissible approach for controlling the laparoscope holder robot. Nevertheless, those results are unsustainable in noisy environments. In those cases, human beings tends to use also visual information in order to filter speech through lip reading. In fact, it has been considered that to observe the speaker is equivalent to a 15 dB gain in the signal to noise ratio [7], [8].

In our previus work [9], two different approaches for solving the laparoscope command problem were presented. The first one was a Gesture Based Command System, which used a set of mouth movements in order to identify the gesture commands using a state machine. The second one was an only Audio Command System, which used 10 english words in order to fit a state machine.

Audio-visual speech recognition (AVSR) has arisen as an alternative when noise or distortion affect the speech [6]. The selection of acoustic features has been studied widely, and the current efforts are concentrated in the extraction of the visual features and the selection of the audio-visual integration model [10], [11], [12]. With the aim of recognizing a little set of commands to handle the three degrees of freedom of the da Vinci's laparoscope holder, an audio-visual speech recognition system is proposed in this work.

This paper is organized as follows. Section 2 presents the visual features used and the visual feature extraction algorithms. Section 3 describes the model used in the AVSR system. Section 4 shows the experimental tests and results of the system. Conclusions of this work are presented in Section 5.

2 Visual Features

The visual features used in speech recognition can be divided in high level, low level and combined features. Model parameters which define the lip contours are used as high level or shape features [6], [11]. The low level features, or appearance features, are obtained as a result of transformations at pixels level of the mouth region [13], [14] and finally, the combined features mix the shape and the appearance of the mouth concatenating the features or using statistical models [15]. Generally, the visual features vector captures dynamic information including the first and second time derivatives. In addition, because the sampling frequency of the audio is higher that the one of the video, the visual features must be interpolated [15].

The MPEG-4 standard has arisen due to the necessity to standardize the virtual objects of real and synthetic video. It includes video codification, geometric compression and audio-video synchronization. This standard presents a complex set of Face Definition Parameters (FDPs) which are used for face standardization, and another set which allows the animation of synthetic face models called Face Animation Parameters (FAPs). FAPs serve to describe face movements (model deformations) with respect to the neutral state face model.

The MPEG-4 standard defines 68 FAPs divided in 10 groups, where groups 2 and 8 are used in speech recognition. They describe the movements of inner and outer lip contour, respectively. For visual speech synthesis, Group 1 is used. It defines 14 clearly distinguishables visemes. A viseme is the visual reference pattern of a phoneme, and it can represent to more than one phoneme.

The FAPs are mesured in specific units called FAPUs (Face Animation Parameter Units) [16]. Figure 1 shows the standardized anthropometric measures. The five FAPU represent the distance between the eyes (ES0), the diameter of the iris (IRISD0), the separation between the eyes and the nose (ENS0), the separation between the mouth and the nose (MNS0), and the mouth width (MW0).

In order to extract the high level visual features of the speech, it is necessary to do precise mouth tracking in the video sequences. Lip tracking is still an open subject in artificial vision due to shape, color and texture complexity, and also because of unexpected changes in illumination [17]. This topic has been successfully treated for lateral face views using controlled background and wearing lipstick, but not with frontal views and without lip markers.

For this work an assisted lip tracking algorithm based on appearance and morphologic restrictions defined in the MPEG-4 standard was designed and implemented. The algorithm assumes that all video frames show frontal face views and that speakers do not uses lip markers. Since psychological studies suggest that the most influent visual feature in lip reading is the outer lip contour, only FAPs from Group 8 were tracked (Figure 2). Moreover, in [11] the authors show that using the Group 2, which describes the inner lip contour, does not increase significantly the performance of the automatic recognition speech system, and

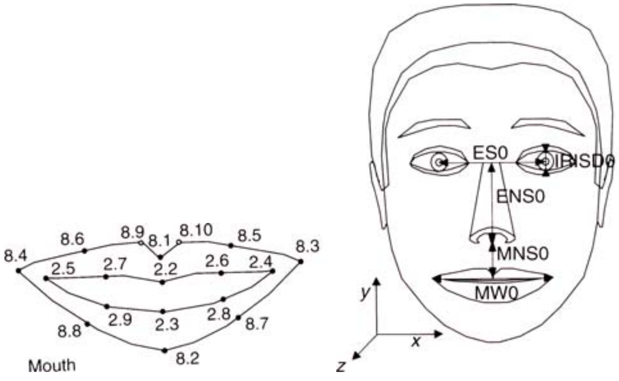


Fig. 1. Groups 2 and 8 of FAPs and the FAPUs measured in a neutral face model

the algorithms are significantly more expensive than those used in outer lip contour tracking.

For calculating the FAPs both magnitude and direction of movement must be preserved, and therefore, they are codified using signed distance functions. Those displacements are standardized using mouth width as normalization factor, which is the FAPU (MW0) for Groups 2 and 8.

Another feature used in this work is mouth roundness. Roundness is found using the Equation 1, in which A corresponds to the area within the outer contour, and d represents the greatest diameter of the mouth region and is equivalent to the mouth width.

$$R = \frac{4A}{\pi d^2} \tag{1}$$

The area is calculated in polar form according to Equation 2, where r_i represents the distance from each one of the 10 points to the mouth center, and $\Delta\Theta_i$ represents the separation angle in between each pair of neighboring points counter clockwise, as shown in Figure 2.

$$A = \sum_{i=1}^{10} r_i^2 \Delta\Theta_i \tag{2}$$

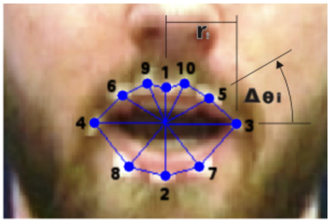


Fig. 2. Outer lip contour defined by the group 8 of the MPEG-4 standard

3 Model Selection

The most popular methods on Automatic Speech Recognition Systems (ASRs) are those based on Hidden Markov Models (HMMs). The HMMs are statistical models whose output is a sequence of symbols. The HMMs deal with the audio sequence as a piecewise statical signal [18], and proved to be more accurate than templates or neural networks at speech recognition [19]. According to recognition task, systems can be classified in the following types: isolated word recognition, where words are separated by pauses; keyword recognition, in which system recognizes certain words in continuous speech; and finally, connected or continuous speech recognition, where the input signal is decoded in a sequence of words, having acknowledged that words are not separated by pauses [20].

HMMs can use either phonemes or words as basic units. There is not direct way to define the number of states for each model, but it has been assumed that using phonemes, three active states is enough [21]. When the models represent words, the model architecture must be assumed in advance. Several configurations must be tested for each word because the system performance strongly depends on the number of states and the probability function of each state.

Figure 3 shows the block diagram of the audio-visual speech recognition system used in our experiments. In this work we used the isolated word recognition approach and words as basic units; we varied the number of states from 3 to 20 active states and used one, two an three probability function of each state. We did not get better results than those obtained using 20 states and one Gaussian per state. We also used the early integration model [10], where the set of the combined visual features from the lip tracking and the audio features is used as the system input.

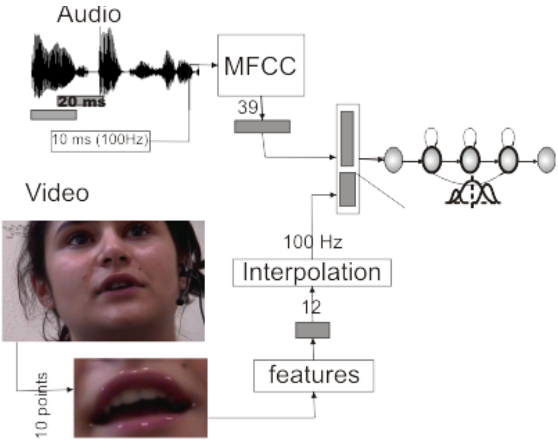


Fig. 3. Block diagram of the AVSR system used in this work

4 Tests and Results

Acquired video data used in this work complies with NTSC standard, whose sampling frequency is of 29.97 frames per second (30 Hertz approximately). Data was recorded in a not controlled enviroment, simulating a realistic situation. There was presence of normal acustic noise sources as computers or other devices. Besides, in the images there was presence of shadows and the light was not controlled. Audio features were extracted using 20 ms windows with overlaps of 50 % between them, which corresponds to 100 Hertz frequency. In order to achieve audio-video synchronization, video features were interpolated from 30 Hertz to 300 Hertz and then subsampled to 100 Hertz.

Principal Components Analysis (PCA) of the FAPs was performed in order to reduce the number of visual features for the audio visual speech recognition system. At the end, only the first three components of the PCA were used, along with the roundness of the region of the mouth, in the visual feature set. In order to include dynamic information, the first two time derivatives of the visual features were also fed to the system.

The system was trained to recognize six spanish words as commands: "izquierda", "derecha", "arriba", "abajo", "adelante" and "atrás". Words' time fetures are shown in Table 1. Video sequences were acquired from 18 people who all were born in Colombia - 5 women and 13 men.

Table 1. Commands used in the experimets

	Derecha	Izquierda	Adelante	Atrás	Arriba	Abajo
mean (seconds)	0.95	1.05	1.11	0.96	0.90	0.96
standard deviation (seconds)	0.19	0.19	0.24	0.18	0.25	0.27

70% of the data was used to train the system, while the remaining 30% was used for testing. In Table 2 test Word Rate Recognition (WRR) is shown, for the cases in which audio, visual and audio-visual features were taken into account. The best performance was obtained with audio features.

Table 2. Word Rate Recognition using 10 and 20 states

audio	97.70	98.85
video	31.03	5.63
audio + video	90.54	97.30

In order to measure the system roboustness against noise, the audio signal was contaminated with white Gaussian noise. The tests were made to match SNR levels between 20 dB and 0 dB. In Figure 4 it can be seen that the performance of audio only system falls abruptly when the noise is as low as 1:100. Also, performance of the audio-visual system was showed superior for all the SNR levels.

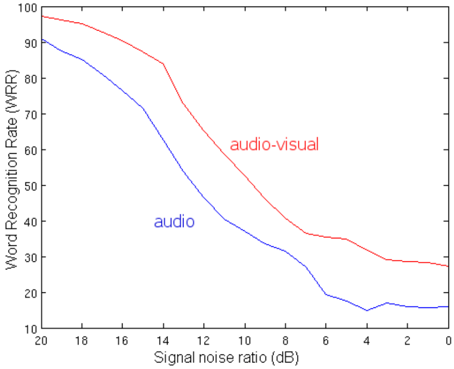


Fig. 4. Audio only and audio-visual WRR vs several SNR levels

5 Conclusion

In this paper we present a speech recognition system for solving the laparoscope command problem. We used audio only, visual only and audio-visual features. Visual features related to mouth shape proved not to be sufficient for solving the recognition task by themselves, but helped when acoustic noise is present in the audio-visual signal. Audio-visual performances exhibited higher errors than the voice based approach when no noise was added, but outperformed in all other cases.

The ASR system based in HMMs using words as basic units in isolated word recognition scheme, which uses both the MFCC as acoustic features and high level visual features based on the standard MPEG-4, presented a WRR near to 100% for recognizing the six spanish words selected as commands. Therefore, the system is reliable for solving the laparoscope holder command task.

Acknowledgments. The authors thank the CHU de Lyon and Doctor Olivier Jegaden for making it possible to access the DaVinci command console.

References

1. Sackier, J., Wang, Y.: Robotically assisted laparoscopic surgery from concept to development. Surgical Endoscopy 8(1), 63–66 (1994)
2. Allen, T.P.K., Goldman, R., Hogle, N.J., Fowler, D.L.: In vivo pan/tilt endoscope with integrated light source, zoom and auto-focusing. Studies in Health Technologies and Informatics, 132–174 (2008)
3. Allaf, M., Jackman, S., Schulam, P., Cadeddu, J., Lee, B., Moore, R., Kavoussi, L.: Voice vs foot pedal interfaces for control of the AESOP robot. Surgical Endoscopy 12, 1415–1418 (1998)
4. Murioz, V., Thorbeck, C.V., DeGabriel, J., Lozano, J., Sanchez-Badajoz, E., Garcia-Cerezoand, A., Toscano, R., Jimenez-Garrido, A.: A medical robotic assistant for minimally invasive surgery. In: IEEE Int. Conf. Robotics and Automation, San Francisco, CA, USA, pp. 2901–2906 (2000)
5. Krupa, A., Gangloff, J., Doignon, C., de Mathelin, M.F., Morel, G., Leroy, J., Soler, L., Marescaux, J.: Autonomous 3-D Positioning of Surgical Instruments in Robotized Laparoscopic Surgery Using Visual Servoing. IEEE transactions on robotics and automation 19(5), 842–853 (2003)
6. Goecke, R.: Current trends in joint audio-video signal processing: A review. In: Eighth International Symposium on Signal Processing and Its Applications (ISSPA 2005), vol. 1, pp. 70–73 (2005)
7. Campbell, R.: Audio-visual speech processing, pp. 562–569. Elsevier, Amsterdam (2006)
8. Campbell, R.: The processing of audio-visual speech: empirical and neural bases. Philosophical Transactions of The Royal Society B 363, 1001–1010 (2008)
9. Gómez, J.B., Ceballos, A., Prieto, F., Redarce, T.: Mouth Gesture and Voice Command Based Robot Command Interface. In: Proceedings of 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), pp. 333–338 (2009)

10. Nefian, A.V., Liang, L., Pi, X., Liu, X., Murphy, K.: Dynamic bayesian networks for audio-visual speech recognition. EURASIP Journal on Applied Signal Processing, 1–15 (2002)
11. Aleksic, P.S., Katsaggelos, A.K.: Comparision of MPEG-4 facial animation parameter groups with respect to audio-visual speech recognition performance. In: IEEE International Conference on Image Processing, ICIP 2005, vol. 3, p. III-501-4 (2005)
12. Kratt, J., Metze, F., Stiefelhagen, R., Waibel, A.: Large vocabulary audio-visual speech recognition using the janus speech recognition toolkit. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 488–495. Springer, Heidelberg (2004)
13. Myung, K., Joung, R., Eun, K.: Speech Recognition with Multi-modal Features Based on Neural Networks. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4233, pp. 489–498. Springer, Heidelberg (2006)
14. Huang, J., Potamianos, G., Connell, J., Neti, C.: Audio-visual speech recognition using an infrared headset. Speech Communication 44, 83–96 (2004)
15. Potamianos, G.: Speech recognition, audio-visual, pp. 800–805. Elsevier, Amsterdam (2006)
16. ISO/IEC: Information technology-generic coding of audio-visual objects, part 2: Visual, ISO/IEC FDIS 14496-2 (final drafts international standard), ISO/IEC JTC1/SC29/WG11 N2502 (1998)
17. Zhilin, W., Aleksic, P., Katsaggelos, A.: Lip tracking for MPEG-4 facial animation. In: Fourth IEEE International Conference on Multimodal Interfaces Processing, vol. 1, pp. 293–298 (2002)
18. Elliot, R.J., Aggoun, L., Moore, J.B.: Applications of mathematics. In: Karatzas, I., Yor, M. (eds.) Hidden Markov Models. Estimation and Control. Springer, New York (1995)
19. Anderson, S., Kewley-Port, D.: Evaluation of speech recognizers for speech training applications. IEEE Transactions on Speech and Audio Processing 3(4), 229–241 (1995)
20. Pasamontes, J.C.: Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en espanol. Estudios de Lingüistica Española (2001)
21. Aguilar, R.C.: Diseño y manipulación de modelos ocultos de markov, utilizando herramientas HTK. Ingeniare. Revista chilena de ingeniería 15(1), 18–26 (2007)

A Rapidly Trainable and Global Illumination Invariant Object Detection System

Sri-Kaushik Pavani[1,2], David Delgado-Gomez[1,2], and Alejandro F. Frangi[1,2,3,⋆]

[1] Research Group for Computational Imaging & Simulation Technologies in Biomedicine, Universitat Pompeu Fabra, Barcelona, Spain
[2] Networking Research Center on Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Spain
[3] Catalan Institution for Research and Advanced Studies (ICREA), Spain
{kaushik.pavani,david.delgado,alejandro.frangi}@upf.edu

Abstract. This paper addresses the main difficulty in adopting Viola-Jones-type object detection systems: their training time. Large training times are the result of having to repeatedly evaluate thousands of Haar-like features (HFs) in a database of object and clutter class images. The proposed object detector is fast to train mainly because of three reasons. Firstly, classifiers that exploit a clutter (non-object) model are used to build the object detector and, hence, they do not need to evaluate clutter images during training. Secondly, the redundant HFs are heuristically pre-eliminated from the feature pool to obtain a small set of independent features. Thirdly, classifiers that have fewer parameters to be optimized are used to build the object detector. As a result, they are faster to train than their traditional counterparts. Apart from faster training, an additional advantage of the proposed detector is that its output is invariant to global illumination changes. Our results indicate that if the object class does not exhibit substantial intra-class variation, then the proposed method can be used to build accurate and real-time object detectors whose training time is in the order of seconds. The quick training and testing speed of the proposed system makes it ideal for use in content-based image retrieval applications.

1 Introduction

Although object detectors based on Haar-like features (HFs) [6] achieve high accuracy rates in real-time [13], training them is a time-consuming task. This is because thousands of weak classifiers based on HFs need to be trained using a database of object and clutter (non-object) images. VJ reported training time in the order of weeks using 180, 000 features on a 466 MHz AlphaStation XP900 [13]. Reduced training time of about 2 days using approximately 20, 000 features can be achieved using the implementation in the OpenCV library [1] on a 3 GHz

[⋆] This work was partially funded by grant TEC2006-03617/TCM, from the Spanish Ministry of Innovation & Science, and grants FIT-360000-2006-55 and FIT-360005-2007-9 from the Spanish Ministry of Industry.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 877–884, 2009.

processor. Though at first glance, it may seem that two days of training time is affordable, the total algorithmic development time generally exceeds this time frame. Many trials may be required to optimize the performance of the detector, which could prolong the effective development time to months. As McCane and Novins [5] point out, long training times make testing new algorithms or verifying past results extremely difficult.

Several possible approaches have been proposed to reduce the high training time. For instance, Wu *et al.* [14] who achieved a reduction in training time of approximately two orders of magnitude by pre-training weak classifiers before the iterative classifier selection procedure. Stojmenovic [12] proposed to reduce the training time by pre-eliminating HFs from the original training set. They eliminate HFs which produce error greater than a pre-determined threshold value. On a database of images containing back-view of Honda Accord cars, they could eliminate 97% of the original features, thereby achieving a potential speed increase of up to two orders of magnitude. However, it is not clear what percentage of HFs can be removed on more challenging images like those of human frontal faces. Pham *et al.* [7] proposed decreasing the training time by pre-computing the global statistics of face and non-face images. They reported a training time of 5 hours and 30 minutes while achieving high accuracy.

In this work, we propose a novel algorithm that reduces the training time to the order of seconds in a conventional desktop computer with a 3 GHz processor. The high training speed is due to the following three reasons. Firstly, a clutter model is used instead of using clutter class images. This results in a substantial reduction of training time because approximately 10^7 clutter image regions are used for training by traditional training methods. The weak classifiers used in the prosed approach, as will be seen in Section 2.1, implicitly incorporate the clutter model and therefore, the model need not be trained. Secondly, we heuristically pre-eliminate HFs in the feature pool to obtain a set of features that make independent measurements on clutter. Using lesser HFs during training also contributes to the faster training speed. Further, the weak classifiers used in our procedure have fewer parameters to be optimized and therefore, are faster to train.

2 Haar-Like Features and Weak Classifiers

Haar-like features (HFs), shown in Fig. 1, are an over-complete set of two-dimensional Haar functions, which can be used to encode local appearance of

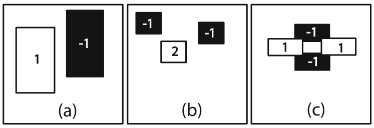


Fig. 1. Typical two-, three- and four-rectangle Haar-like features. The numbers shown on the rectangles refer to the weights assigned to each of them.

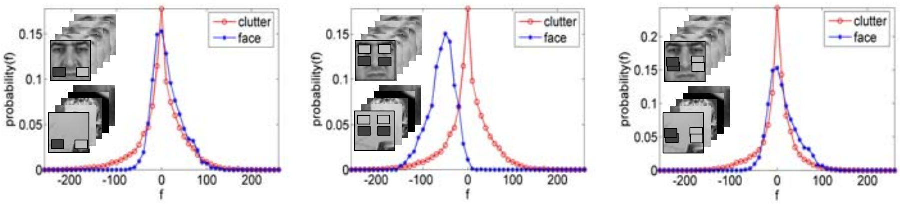


Fig. 2. Three histograms of feature values obtained by evaluating face and clutter class images on HFs are shown. To the left of the histograms, the HFs that were used for evaluation have been super-imposed on a typical training image. As Huang and Mumford [3] observed, the distribution of feature values from clutter images tends to a Laplacian distribution centered at zero.

objects [6]. The feature value f of a Haar-like feature which has k rectangles is obtained as in (1). The quantity $\mu^{(i)}$ is the mean intensity of the pixels in image $\mathbf{x}$ enclosed by the i^{th} rectangle and $w^{(i)}$ is the weight assigned to the i^{th} rectangle. The weights assigned to the rectangles of a HF are set to default numbers satisfying (2). Weak classifiers that label an image $\mathbf{x}$ as object (+1) or clutter (-1) can be expressed as in (3). The quantity $\theta \in \Re$ is a threshold value, and $p \in \{1, -1\}$ can be used to invert the inequality relationship. Training such a weak classifier involves setting appropriate values to its threshold and polarity coefficients (θ^*, p^*) such that the overall error is minimized. Formally, $[\theta^*, p^*] = \arg\min_{[\theta,p]} \sum_{i=0}^{n_o+n_c} \epsilon^{(i)}$. If a training image is correctly classified, then its error is $z^{(i)}$, else it is 0. The term $z^{(i)}$ is the weight assigned to the training image $\mathbf{x}^{(i)}$. The quantities n_o and n_c are the number of object and clutter class training images, respectively. Training the weak classifiers as in (3) can be intuitively understood from Fig. 2. For each HF shown in Fig. 2, histograms of the feature value, f, have been obtained from object (human frontal face) and clutter training images. During training, θ is set to the value of f that best separates object and clutter examples.

$$f = \sum_{i=1}^{k} w^{(i)} \cdot \mu^{(i)} \qquad (1)$$

$$\sum_{i=1}^{k} w^{(i)} = 0 \qquad (2)$$

$$h(\mathbf{x}) = \begin{cases} +1, f_{(\theta,p)} > 0 \\ -1, \text{otherwise} \end{cases} \qquad (3)$$

$$f_{(\theta,p)} = (f - \theta) \cdot p \qquad (4)$$

2.1 A Clutter Model

When a HF is evaluated on a clutter image, the expectation value of the output can be expressed as in (5).

$$E(f) = E\left(\sum_{i=1}^{k} w^{(i)}\mu^{(i)}\right) = \sum_{i=1}^{k} w^{(i)} E\left(\mu^{(i)}\right) \tag{5}$$

The clutter class, being generic, may contain any image with any appearance pattern. Effectively, every pixel of a generic clutter image is a random variable which can take any value between the minimum and the maximum permitted pixel values in an image representation (N_{min} and N_{max}) with equal probability. For example, in gray-level images, $N_{min} = 0$ and $N_{max} = 255$. Therefore, the expected value of mean of pixel values within any rectangular region, $E(\mu) = 0.5(N_{max} + N_{min})$. Rewriting (5) using (2), we get (6).

$$E(f) = 0.5(N_{max} + N_{min}) \sum_{i=1}^{k} w^{(i)} = 0 \tag{6}$$

Therefore, the probability that the feature value of a HF on a clutter image to be greater than (or lesser than) 0 is 0.5. Mathematically, $\mathbb{P}(f \cdot p > 0 | \mathbf{x}^{(i)} \in \texttt{Clutter}) = 0.5$. Using the terminology introduced in (4),

$$\mathbb{P}(f_{(0,p)} > 0 | \mathbf{x}^{(i)} \in \texttt{Clutter}) = 0.5 \tag{7}$$

The clutter model in (7) can be observed from the clutter histograms shown in Fig. 2. Note that the clutter histograms are all symmetric and centered at $f = 0$.

2.2 Proposed Weak Classifier

The proposed weak classifier utilizes the clutter model in (7) by setting its threshold $\theta = 0$ so that it labels 50% of the clutter correctly. Since θ is already set, training the proposed weak classifier only involves setting an appropriate value to the polarity term (p^*) such that the training error is minimized as shown in (9). As θ need not be optimized, the training speed of the weak classifiers is much higher than the traditional ones as in (3).

$$h(\mathbf{x}) = \begin{cases} +1, & f_{(0,p)} > 0 \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

$$p^* = \arg \min_{p \in \{1,-1\}} \sum_{i=0}^{n_o} \epsilon^{(i)} \tag{9}$$

The object detectors are built by arranging weak classifiers as in (8) according to the rejection cascade architecture [2]. This architecture has been preferred for building object detectors as it is conducive for fast scanning of an image [13]. A rejection cascade, as illustrated in Fig. 3, consists of multiple nodes connected in series. Each node is a binary classifier that classifies an input sub-region as object or clutter. Each node consists of multiple weak classifiers which are

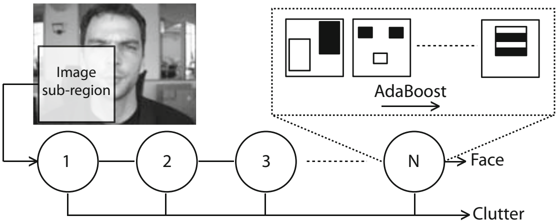


Fig. 3. A cascaded classifier consists of multiple nodes arranged in a degenerated decision tree fashion. An input image is scanned at different scales and positions for the presence of a face. If an image sub-region is classified as a face by all the sub-regions of the face, then it is labeled a face.

selected iteratively using the AdaBoost procedure [10]. The weighted decision of all the weak classifiers in a node is output as the decision of the node.

2.3 Pre-eliminating Redundant HFs

As mentioned before, HFs are an over-complete set of features, therefore, they are redundant. Conventional object detectors avoid selecting redundant features in different nodes by training each node with bootstrapped set of clutter images [13]. In other words, features selected for different nodes are suitable for classifying different subsets of clutter images. In our case, since clutter images are not used, the over-complete set of HFs need to be pruned heuristically after each node is built so that neither the previously selected features nor similar ones are selected again. Similarity between two HFs is measured by the amount of overlap between its rectangles. For example, the HFs illustrated in Fig. 2(left) and Fig. 2(middle) do not overlap at all, therefore, they are considered to make independent measurements on a clutter image. On the contrary, the HFs illustrated in Fig. 2(middle) and Fig. 2(right) have more than 50% overlap, and therefore they are considered to make redundant measurements. To build the proposed object detector, we generated a feature pool with $7,200$ HFs in which no HF in the feature pool has more than 50% overlap with the rest of the features.

3 Experimental Setup and Results

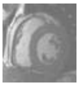
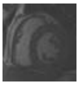
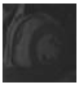
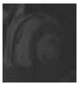
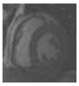
The proposed weak classifiers described above were trained for two very different object detection problems: detection of human frontal faces in photographs and detection of the human heart in short-axis cardiac Magnetic Resonance Images (MRI). For this purpose, two object databases (`face` and `heart`) were used. The `face` database was composed of 5000 images. The faces in this database exhibit an out-of-plane rotation of up to $\pm 10^\circ$ and various expressions. The `heart` database consisted of 493 short-axis MR heart images. In comparison to the images in the `face` database, the images in the `heart` database exhibit less

Table 1. Comparison of training time

Method	Number of features in the feature pool	Number of classifiers trained	Number of object images used	CPU speed (GHz)	Training time
Proposed (Face)*	7,800	3,200	5,000	3.0	96s
VJ [13]	40,000	4,297	9,500	0.4	weeks
LZZBZS [4]	n/a	6,000	2,546	0.7	weeks
WBMR [14]	40,000	3,870	5,000	2.8	13h20m
PC [7]	295,920	3,502	5,000	2.8	5h30m
Proposed (Heart)*	7,800	1,000	493	3.0	30s
VJ (Heart)*	180,000	300	493	3.0	22h

* Results from our implementation.

Table 2. True positive rate in simulated test datasets

Method	DS1^a	DS2^b	DS3^c	DS4^d	DS5^e	DS6^f	DS7^g	DS8^h
Proposed (Face)*†	88.0	87.4	86.5	88.0	88.0	88.0	88.0	84.7
VJ (Face)*	90.3	90.3	90.3	87.4	86.2	87.0	83.9	80.2
VJ (Face)*†	90.3	85.1	72.5	87.4	78.7	84.0	81.2	60.5
								
Proposed (Heart)*†	97.3	97.3	94.6	97.3	97.3	97.3	96.7	93.8
VJ (Heart)*	98.7	98.7	98.7	90.3	85.2	96.8	93.0	76.3
VJ (Heart)*†	98.7	81.2	63.2	90.3	20.3	69.6	35.2	0.0

* Results from our implementation. † Results without variance normalization.
^{a} DS1: Original test images. ^{b} DS2: Intensity values are globally divided by 2.
^{c} DS3: Intensity values are globally divided by 3. ^{d} DS4: Histogram equalized images.
^{e} DS5: Gamma corrected image ($\gamma = 0.8$). ^{f} DS6: Gamma corrected image ($\gamma = 0.9$).
^{g} DS7: Gamma corrected image ($\gamma = 1.1$). ^{h} DS8: Gamma corrected image ($\gamma = 1.2$).

intra-class appearance variation. The face detectors were tested on MIT+CMU frontal face database [9]. The heart detectors were tested on a set of 293 images. The speed of training of the face and heart detectors, in comparison to other methods, is tabulated in Table 1.

We tested the accuracy of the object detectors by transforming the test images artificially to simulate global illumination changes. On each of the transformed database, the accuracy of the VJ-type detector and the proposed method were measured and the results are tabulated in Table 2. We observed that, in

Table 3. Comparison of accuracy of the face and heart detectors

Method	Face		Heart	
	FD [a]	TPR [b]	FD [a]	TPR [b]
Proposed	912*	88.0*	2*	97.3*
VJ [13]	95	90.8	2*	98.7*
LZZBZS [4]	90	92.5	n/a	n/a
WBMR [14]	85	92.5	n/a	n/a
PC [7]	100	90.0	n/a	n/a
RBK [9]✠	95	89.2	n/a	n/a
SK [11]✠	65	94.5	n/a	n/a
RYA [8]✠	78	94.8	n/a	n/a

[a] FD: Number of false detections. [b] TPR: True positive rate.
* Results from our implementation. ✠ Methods not based on HFs.

contrast to VJ detector, the proposed detector performed consistently to all the monotonic image transformations applied to the test images. This is because, the detector uses weak classifiers that make decision based on the sign of the feature value of a HF, and not based on the magnitude of the feature value of the HF. In theory, the accuracy of the proposed detector should not change if any monotonic transformations are applied to images. However, we see that the accuracy decreases in DS3 and DS8. This is because, two image patches (with different original intensities) might end up have the same average intensities after image transformation, and therefore, not satisfy (3) because of saturation of intensity values (as in the case of DS8) or because of rounding errors in the division process (as in the case of DS3). The results of the VJ-type detector with and without variance normalization are also tabulated in Table 2. The proposed detector does not require variance normalization procedure as the sign of the feature value of any HF is not affected by the variance normalization process. Thus, the computation of the integral image square and the computation of standard deviation of each image sub-region can be avoided during the detection process, which adds to the speed of detection. The time required to process all the images in the test set by the face and the heart detectors were 41s and 12s. This includes the time to read the image, computation of integral image(s), the scanning, and the clustering process to merge multiple detections. Our implementation of VJ procedure (with variance normalization) took 62s and 14s, respectively. The testing times were measured on a 3 GHz CPU.

The number of false detection by the the face and the heart detectors, along with the state-of-the-art methods, is listed in Table 3. The face detector achieved a false positive rate of 9.2×10^{-5} (912 false detections), which is approximately 10 times worse than the state-of-the-art detectors. However, the number of false detections by the heart detector was only 2, which represented a false positive rate of 3.3×10^{-6}.

4 Conclusions

In this paper, we have presented a novel training procedure for object detection systems and compared its performance, both during training and testing phases, with the state-of-the-art techniques. The advantages of adopting proposed technique include fast training in the order of seconds, global illumination invariance and real-time detection speed. The disadvantage of this method is that it produces more false positives with respect to the state-of-the-art.

The quick training and testing speed of the proposed technique makes it ideal for content based image retrieval systems - where a user makes a query (an image patch), and asks the system to automatically find similar patches in a huge database of images. The existing methods, by the virtue of being slow to train, cannot be used in such scenarios.

References

1. OpenCV library, http://sourceforge.net/projects/opencvlibrary/
2. Baker, S., Nayar, S.K.: Pattern rejection. In: CVPR 1996, pp. 544–549 (1996)
3. Huang, J., Mumford, D.: Statistics of natural images and models. In: CVPR 1999, pp. 541–547 (1999)
4. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
5. McCane, B., Novins, K.: On training cascade face detectors. In: IVCNZ 2003, pp. 239–244 (2003)
6. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: ICCV 1998, pp. 555–562 (1998)
7. Pham, M.-T., Cham, T.-J.: Fast training and selection of Haar features using statistics in boosting-based face detection. In: ICCV 2007, pp. 1–7 (2007)
8. Roth, D., Yang, M., Ahuja, N.: A SNoW-based face detector. In: NIPS 2000, pp. 855–861 (2000)
9. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE TPAMI 20(1), 23–38 (1998)
10. Schapire, R.E.: A brief introduction to boosting. In: IJCAI 1999, pp. 1401–1406 (1999)
11. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: CVPR 2000, pp. 746–751 (2000)
12. Stojmenovic, M.: Pre-eliminating features for fast training in real time object detection in images with a novel variant of AdaBoost. In: CIS 2006, pp. 1–6 (2006)
13. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)
14. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE TPAMI 30(3), 369–382 (2008)

Expanding Irregular Graph Pyramid for an Approaching Object*

Luis A. Mateos, Dan Shao, and Walter G. Kropatsch

Vienna University of Technology,
Pattern Recognition and Image Processing Group,
Favoritenstr. 9/1832, A-1040 Vienna, AUSTRIA
{lam,shao,krw}@prip.tuwien.ac.at
http://www.prip.tuwien.ac.at/

Abstract. This paper focus on one of the major problems in model-based object tracking, the problem of how to dynamically update the model to adapt changes in the structure and appearance of the target object. We adopt Irregular Graph Pyramids to hierarchically represent the topological structure of a rigid moving object with multiresolution, making it possible to add new details observed from an approaching object by expanding the pyramid.

Keywords: irregular graph pyramid; adaptive representation; object structure; tracking model.

1 Introduction

One of the major problems for model-based object tracking is: how to dynamically update the model to accommodate the object appearance and structure changes due to the changes in the surrounding conditions against which the tracked object is observed [1].

By reviewing the previous works for model updating, a method for adjusting features while tracking is presented by Collins and Liu [2]. Their hypothesis is that best discriminating features between object and background are also best for tracking the object. This method uses the immediate previous frame as the training frame and the current as the test frame for the foreground and background classification. In [3] the template is first updated with the image at the current template location. To eliminate drift, this updated template is then aligned with the first template to give the final update.

However, these methods update their models by getting new information of the target object from resampled input image frames, with a fixed resolution.

The properties of the Irregular Graph Pyramid and its applications in image processing motivated us to use the advantages of its topological/structural hierarchy features in tracking. With its hierarchical feature, we are able to represent the moving object in multi resolution, and incrementally update it by adding new details from the approaching target object.

* Supported by the Austrian Science Fund under grant P18716-N13.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 885–891, 2009.

Each level of the pyramid is represented by a graph which embeds the topological structure of the object at certain resolution. Such graphs are built of vertices and spatial edges. In the vertex; attributes like size, color and position of the corresponding pixels (region) can be stored. The spatial edges are used to specify the spatial relationships (adjacency, border) between the vertex (regions). Tracking methods using structural information often employ graphs to describe the object structure. Locally, features describe the object details; globally, the relations between features encode the object structure.

By exploiting the spatial and temporal structure of the scene [4], Artner and Lopez improve the performance of object tracking in video sequences. They present a structural tracking approach which is robust to different types of occlusions, as the structural information is able to differ objects by their different structural types.

Gomila and Meyer [5] represent as a region adjacency graph each image of a video sequence. For object tracking a graph matching is performed, in which the intrinsic complexity of graph matching is greatly reduced by coupling it with segmentation.

Less work has been done for tracking with graph pyramid. [6] presents a method of tracking objects through occlusions using graph pyramid. They apply graph matching to find the matching between the vertices in current image frame with the ones in previous image frames. Instead of doing the graph matching which is known as computationally expensive, we propose a top - down pyramid reconstruction approach to avoid graph matching.

Major contributions of this paper are 1) Our model hierarchically represents a moving object with multi resolution; 2) For an approaching object, we encode the new details by expanding the pyramid structure; 3) Compared to other pyramid tracking methods, we use a top - down pyramid reconstruction approach instead of bottom-up recomputing the graph pyramid for each frame. In such way, computational costs would be reduced.

Organization of paper. In section 2 we recall the concept of Irregular Graph Pyramid (IGP). In section 3 we describe the process of tracking with IGP. Section 4 describes a concept of Adatpive Zoom in for tracking the approaching object. Section 5 finalizes with conclusion and open questions.

2 Recall of Irregular Graph Pyramid

An irregular pyramid combines graph structures with hierarchies. Each level is a graph describing the image with various resolutions by contracting the graph from the level below. Specifically, contraction is a process to take the attributes of all children as input and then compute the parent's attribute as output, removes the edges from input graph while simultaneously merging together the vertices it used to connect [7].

For this paper we are considering combinatorial maps. A combinatorial map is a topological model which allows to represent subdivided objects as planar graphs. A 2D combinatorial map is defined by a triplet $M = (D, \sigma, \alpha)$ where D is a finite set of darts, σ is a permutation on D and α is an involution on D without fixed point [8]. For each dart, σ gives the next dart by turning around the vertex v in the positive orientation (clockwise); For each dart, α gives the other dart of the same edge e. There are always two darts corresponding to a same edge, α allows to retrieve edge e, and σ allows to retrieve the vertex v.

Taking a simplified image of a cup as example, we build the base graph as the input image, where each vertex represent a pixel in the input image. Then use the contraction methods to build the irregular pyramid. Such approach would lead to a pyramidal structures like the following figure:

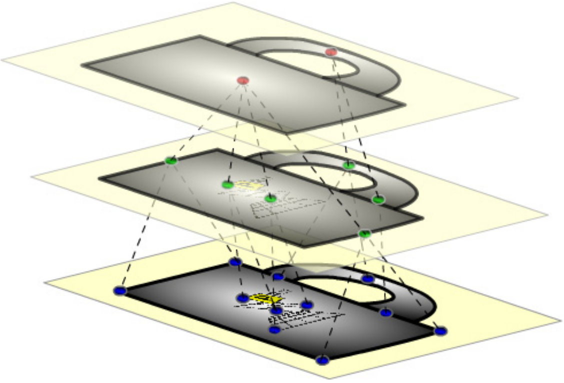


Fig. 1. Irregular Graph Pyramid

Level 0: The base level of the pyramid consists in a geometric description of the underlying image (here a simplified image of a cup).

Level 1: The second level of the pyramid, simpler boundaries are abstracted from base level (like the handle and the logo of the cup).

Level 2: Adjacent parts of the cup are grouped in order to represent compound abstract objects.

In irregular pyramids, during the building process of pyramid, the adaptive contraction of the structure preserves its topological structure.

3 Tracking with Irregular Graph Pyramid

In this paper we only consider tracking linear object movement. Translation, scaling and rotation are the basic types of linear object movement. In this section, we describe how to track the translation movement using graph pyramid. In the following section, we focus on tracking the scaling movement (approaching object). Rotation is not covered by this paper. However, in our other works, we present the concept of topological completion to reconstruct the closed surface of a 3D object for tracking a rotating 3D object.

For initialization, a pyramid is bottom-up built for the target object from a video frame. The target object is represented at different levels of resolution using a graph for each level. We build the base graph as the input image, where each vertex represent a pixel. We save the coordinates of pixels in the attributes of the vertices in base graph. And discriminability can be computed locally similar to Lowe's SIFT descriptor [**?**].

In the contraction process, the vertices with most discriminative feature survive. The coordinates of the surviving vertices are preserved in the attributes of the parents. For the non surviving vertices, coordinates are saved in the contracted edges for the purpose of later pyramid reconstruction.

In such way, the pyramid apex encodes the most discriminative feature of the target object as well the coordinate of the most discriminative feature. For instance, for a cup with yellow logo on the surface, the apex encodes the yellow logo and the coordinates of the logo.

For a rigid object, the structure of the pyramid is invariant to any geometric transformation. By estimating the location of the logo (which is abstracted as the apex), we are able to locate the whole pyramid. All the vertices can be accessed efficiently from the apex by following the parent - children path. The construction of the pyramid is a bottom - up process while the reconstruction from the apex is a top - down process. In such way we can reconstruct the whole pyramid by only locating the apex point.

An example of the tracking process is shown Fig. 2. At frame i the object is detected and its pyramid is initially built. The apex encodes the most discriminative feature of the object (the yellow logo). In the next frames $i + n$ the object moves, we detect the location of the yellow logo by motion estimation method. Once the apex coordinate (coordinate of yellow logo) is known, we can reconstruct the pyramid until we retrieve all the vertices and their coordinates

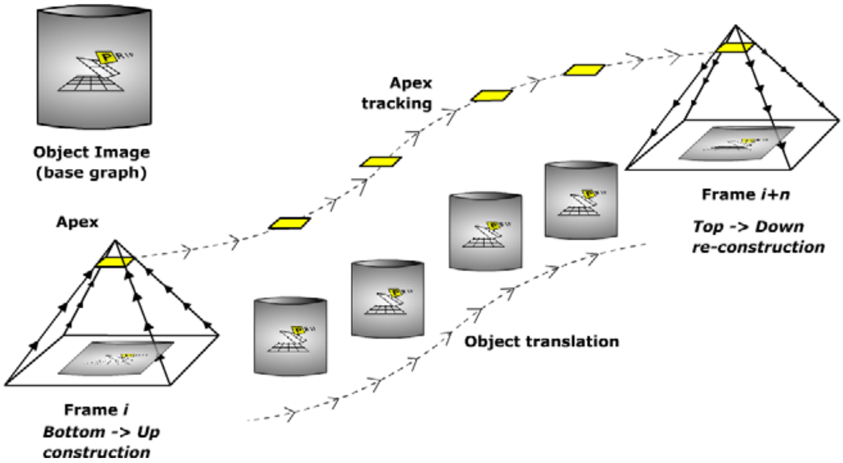


Fig. 2. Tracking with Irregular Graph Pyramid

of base graph. As the base graph of pyramid present the image of the target object, the target object at frame $i + n$ is tracked.

4 Adaptive Zoom - In

Model-based object tracking intend to keep a clear and detailed description of the object by gathering its latest descriptors available, these descriptors describe elementary characteristics of the object such as the color, the shape, the texture, among others. In order to do this, the model must be able to adapt the object changes. If an object changes its details for a given reason then the model must have such a robust adaptive system that keeps a clear identification and location of this tracking object.

Every frame is different, so new relevant pieces of information or features from the object may appear due to different reasons such as appearance or pose changes. We are treating the current frame as the latest source of information, containing the ultimate object descriptors. This new information may include changes in its internal or structural descriptors. Traditional model based representations are reliable and robust in scenes where the object is consistent, no expansion of details, no considerable illumination changes. In this section we present a case and the proposed solution using an adaptive zoom - in; if the object gets closer to the camera, the distance camera-object changes and we will obtain a bigger picture (higher resolution) of the object and more detailed descriptors will appear.

From the irregular graph pyramid perspective, this new image can be seen as a proportional projection of the original base graph which includes more detailed descriptors. The pyramid will expand one level below the current base graph, this new base graph encodes both structures due to a higher resolution of the object. And the problem is to find the vertical connection between the new base graph and the existing upper part of the pyramid.

Let $G = (V, E)$ be the graph at the base of the original pyramid and $\tilde{G} = \left(\tilde{V}, \tilde{E}\right)$ the new base graph.

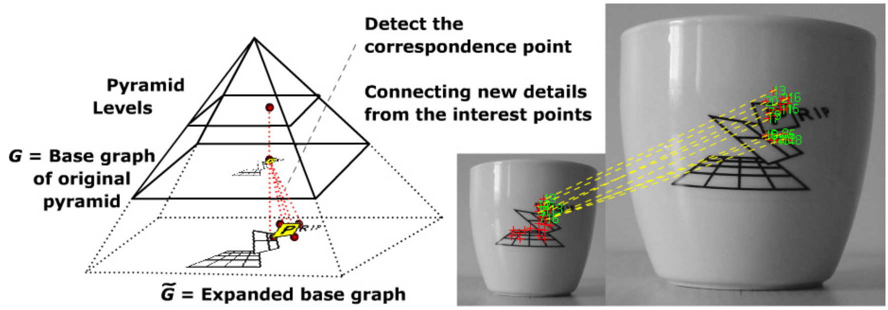

Pyramid Levels
G = Base graph of original pyramid
Detect the correspondence point
Connecting new details from the interest points
$\tilde{G}$ = Expanded base graph

Fig. 3. Pyramid base projection to a lower level

Other than in most parameter optimization tasks, both the vertices of the lower level $\tilde{G}$ and the next upper level G are know as well as the reduction function. The only unknown is the contraction kernel K such that $G = \tilde{G}/K$.

$\tilde{G} = \left(\tilde{V}, \tilde{E}\right)$ corresponds to the image grid with higher resolution, which is considered as a 4-connected graph. For each vertex $v \in \tilde{V}$, a set of allowed candidate parents are computed by finding the closer correspondences. Then, a new parent is chosen from the set of allowed candidate parents. The vertex is relinked to the new parent and the graph structure and attributes of vertices are updated accordingly. This procedure is repeated until a stable configuration is reached. The new parent p_{new} is chosen for each vertex v such that the difference between $g\left(\tilde{v}\right)$ and $g\left(p_{new}\right)$ is minimized. The rules which determine the selection of a new parent for a vertex can be formulated as an energy minimization problem relinking [9]. An energy

$$E = \sum_{v \in \tilde{V}} n(v)\left[g(v) - g(p(v))\right] \tag{1}$$

can be defined. Here, $n(v)$ denotes the area of the receptive field of v, $g(v)$ denotes its gray level or local feature value and $p(v)$ denotes the parent assignment. As pointed out by Nacken [9], this relinking algorithm may destroy the connectivity of the receptive fields. Improve our method with Nacken's modification is one of our future directions.

By graph relinking, we put higher resolution image frame as new base graph, then define contraction kernel that produce the old base. Instead of bottom-up rebuilding pyramid by applying contraction kernel, we use the graph at the base of the original pyramid as fixed upper level graph, and try to define the expansion kernel. In such way, the original pyramid structure remains unchanged, we only attach a lower level base graph into this pyramid.

Assumption. Considering the speed of the moving object, we assume the approaching speed is not too fast. The current size of the target object can not exceed twice as the one in previous image frame, which means the maximum scaling factor can not exceed 2. Otherwise there might be a gap between the new base graph and the old base graph so that we have to insert extra levels to bridge the new base graph with the old base graph.

5 Conclusions

This paper presents a novel concept of using irregular graph pyramid to represent a moving object in multi resolution. This concept can be applied into many tracking applications. In this paper, we focus on how to track approaching object (scaling movement) by expanding the irregular graph pyramid. Considering the computational costs, we propose a top-down pyramid reconstruction approach, instead of bottom - up rebuilding pyramid for the target object in each image frame. Our future work will concentrate on how to extend this tracking framework to track other object movement such as reflection, shear and non linear movement.

References

1. Tang, F., Tao, H.: Probabilistic Object Tracking With Dynamic Attributed Relational Feature Graph. IEEE Trans. Circuits Syst. Video Techn. 18(8), 1064–1074 (2008)
2. Collins, R., Liu, Y., Leordeanu, M.: On-Line Selection of Discriminative Tracking Features. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
3. Matthews, I., Ishikawa, T., Baker, S.: The template Update Problem. IEEE Transaction Pattern Analysis and Machine Intelligence 26(6), 810–815 (2004)
4. Lopez Marmol, S.B., Artner, N.M., Iglesias, M., Kropatsch, W., Clabian, M., Burger, W.: Improving Tracking Using Structure. In: proceedings of Computer Vision Winter Workshop (CVWW 2008), February 4–6, pp. 69–76. Moravske Toplice, Slovenia (2008)
5. Gomila, C., Meyer, F.: Graph-based object tracking. In: ICIP 2003, vol. 2, 3, p. 2. Thomson Inc. - Corporate Res, Princeton (2003)
6. Conte, D., Foggia, P., Jolion, J.-M., Vento, M.: A graph-based, multi-resolution algorithm for tracking objects in presence of occlusions. Pattern Recognition 39(4), 562–572 (2006)
7. Kropatsch, W.G.: Building irregular pyramids by dual-graph contraction. In: IEE Proceedings- Vision Image and Signal Processing, vol. 142(6), pp. 366–374 (1995)
8. Brun, L., Kropatsch, W.G.: Dual Contraction of Combinatorial Maps. Technical Report PRIP-TR-54 Institute f. Computer Aided Automation 183/2, Pattern Recognition and Image Processing Group, TU Wien, Austria (1999a)
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision, Corfu, Greece, September 1999, pp. 1150–1157 (1999)
10. Nacken, P.F.M.: Image segmentation by connectivity preserving relinking in hierarchical graph structures. Pattern Recognition 28(6), 907–920 (1995)

Learning Relational Grammars from Sequences of Actions

Blanca Vargas-Govea and Eduardo F. Morales

National Institute of Astrophysics, Optics and Electronics
Computer Science Department
Luis Enrique Erro 1, 72840 Tonantzintla, México
{blanca,emorales}@inaoep.mx

Abstract. Many tasks can be described by sequences of actions that normally exhibit some form of structure and that can be represented by a grammar. This paper introduces FOSeq, an algorithm that learns grammars from sequences of actions. The sequences are given as low-level traces of readings from sensors that are transformed into a relational representation. Given a transformed sequence, FOSeq identifies frequent sub-sequences of n-items, or n-grams, to generate new grammar rules until no more frequent n-grams can be found. From m sequences of the same task, FOSeq generates m grammars and performs a generalization process over the best grammar to cover most of the sequences. The grammars induced by FOSeq can be used to perform a particular task and to classify new sequences. FOSeq was tested on robot navigation tasks and on gesture recognition with competitive performance against other approaches based on Hidden Markov Models.

1 Introduction

Sequences are used to describe different problems in many fields, such as natural language processing, music, DNA, and gesture recognition, among others. The analysis of such sequences often involves exploiting the information provided by the implicit structure of the sequences that can sometimes be represented by a grammar. Learning grammars from sequences offers several advantages. Suppose that you have a set of sequences of actions performed by a robot to move between two designated places avoiding obstacles. If we could infer a grammar from such sequences, we could use it to recognize when a robot moves between two places and also to generate a sequence of actions to perform such task. Another advantage is that a grammar normally includes sub-concepts that can include other sub-concepts or primitive actions, which can be used to solve other related sub-tasks.

Grammars can be represented by different formalisms, the most commonly used is context-free grammars (CFGs). In this paper, we use Definite Clause Grammars (DCGs), a generalization of CFGs that use a relational representation. This is important as it allows us to apply the learned grammar to different instantiations of a more general problem.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 892–900, 2009.

We focus on learning grammars from sequences of actions that can be used as programs to execute a task and as classifiers. The training sequences are provided by the user, the main idea is to show the system what to do instead of how to do it (e.g., steering the robot avoiding obstacles or moving a hand), simplifying the programming effort. The set of traces consists of low-level sensor readings that are transformed into a high level representation. The transformed sequences are given to an algorithm (FOSeq) that induces grammars that can be used to reproduce the original human-guided traces and to identify new sequences.

The approach was applied in two domains: (i) robot navigation and (ii) gesture recognition. We tested the learned navigation grammars in a robotics scenario with both simulated and real environments and show that the robot is able to accomplish several navigation tasks. The gesture recognition was tested using a public database of gestures, showing that the classification accuracy is competitive with other common approaches with the advantage of learning an understandable representation.

This paper is organized as follows. Section 2 reviews related work. Section 3 describes the grammar learning algorithm. Section 4 presents the navigation task while Section 5 describes the gesture recognition experiment. Conclusions and future research directions are given in Section 6.

2 Related Work

Grammar induction has been commonly studied in the context of Natural Language Processing. Most of the approaches use grammars as parsers and focus on specific problems and are rarely used to solve other related problems. Other researchers have tried to induce grammars using a relational representation. EMILE [1] and ABL [2] are algorithms based on first order logic that learn the grammatical structure of a language from sentences. Both algorithms focus on language and it is not easy to extend them to other applications. GIFT [3] is an algorithm that learns logic programs but it requires an initial rule set given by an expert. In contrast, FOSeq learns relational grammars from sequences of actions, where the grammars are logic programs that can reproduce the task described by the sequences. A grammar induction technique closely related to our work is SEQUITUR [4], an algorithm that infers a hierarchical structure from a sequence of discrete symbols. However, SEQUITUR handles only constant symbols, is based on bi-grams (sets of two consecutive symbols), and consequently, the learned rules can only have pairs of literals in their bodies, and it does not generalize rules. FOSeq employs a relational approach, is not restricted to bi-grams and is able to generalize.

Learning from sequences has been used in robotics to learn skills. In [5] a robot has to learn movements (e.g., aerobic-style movement) showed by a teacher. To encode the movements, and subsequently be able to recognize them, they used Hidden Markov Models (HMM). However, this representation is not easy to interpret and does not capture the hierarchical structure of the sequences. In our approach, hierarchical skills can be learned from sequences of basic skills.

Another domain that has been used to learn from sequences is gesture recognition, which is an important skill for human–computer interaction. Hidden Markov Models and neural networks are standard techniques for gesture recognition. However, most of the approaches have emphasized on improving learning and recognition performance without considering the understandability of the representation [6].

3 Learning Grammars with FOSeq

The general algorithm can be stated as follows: from a set of sequences, (i) learn a grammar for each sequence, (ii) parse all the sequences with each induced grammar, evaluate how well each grammar parses all the traces, and (iii) apply a generalization process to the best grammar trying to cover most of the sequences.

1: Grammar induction. Given a trace of predicates the algorithm looks for n-grams (e.g., sub-sequences of n-items, in our case n-predicates) that appear at least twice in the sequence. As in Apriori [7], the candidate n-grams are incrementally searched by their length. The search starts with $n = 2$ and ends when there are no more repeated n-grams for $n \geq 2$. The n-gram with the highest frequency of each iteration is selected, generating a new grammar rule and replacing in the sequence, all occurrences of the n-gram with a new non-terminal symbol. If there is more than one n-gram with the same highest-frequency, the longest n-gram is selected. If there are several candidates of the same length, the algorithm randomly selects one of them. Repeated items are removed because items represent actions that are executed continuously while specific conditions hold. Therefore, the action will be repeated while its conditions are satisfied even if it appears only once.

Example. Let us illustrate the grammar induction process with the following sequence of constants: S → a b c b c b c b a b c d b e b c. FOSeq looks for n-grams with frequency ≥ 2 as candidates to build a rule. In the first iteration there is only one candidate: {b c} with five appearances in the sequence. This n-gram becomes the body of the new rule R1→b c. The n-gram is replaced by the non-terminal R1 in sequence S, generating S_1 and removing repeated items. In the second iteration FOSeq finds three candidates: {R1 b}, {a R1} and {a R1 b} with two repetitions each. FOSeq selects the longest item: {a R1 b} and a new rule is added: R2→a R1 b. Sequence S_2 does not have repeated items and the process ends. Figure 1 shows the learned grammar where R1 and R2 can be seen as sub-concepts in the sequence.

When the items of the sequence are first–order predicates the learned grammar is a definite clause grammar (DCG). DCGs are an extension of context free grammars that are expressed and executed in Prolog. In this paper we have sequences of predicates that are state-action pairs with the following format: pred1($State_1$,$action_1$), pred2($State_2$,$action_1$), pred1($State_3$,$action_2$), ...

For repeated predicates a new predicate is created, where *State* is the first state of the first predicate and *Action* is the action of the last predicate. For

S_2 → R2 d b e R1
R1 → b c
R2 → a R1 b

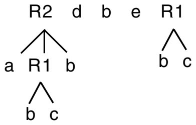


Fig. 1. Induced grammar for sequence S. S_2 is the compressed sequence after the induction process. The grammar describes the hierarchical structure of the sequence.

instance, suppose that the following pair of predicates is repeated several times in the sequence: ..., pred1($State_j$,$Action_1$), pred2($State_k$,$Action_2$), ..., then the following new predicate is created:
newpred($State_1$,$Action_2$) ← pred1($State_1$,$Action_1$), pred2($State_2$,$Action_2$) .

2: Grammar evaluation. A grammar is created for each sequence. Every learned grammar is used to parse all the sequences in the set of traces provided by the user and the best evaluated grammar is selected. The measure of how well the grammar parses is calculated using the following function:

$$eval(g_i) = \sum_{i=1}^{m} \frac{c_i}{c_i + f_i}$$

where g_i is the grammar being evaluated from a set of m sequences, c_i and f_i are the number of items that the grammar is able or unable to parse respectively and i is the index of the sequence being evaluated. When a grammar is not able to parse a symbol, it is skipped. FOSeq selects the grammar that best parses the set of sequences.

3: Generalization. The key idea of the generalization process is to obtain a new grammar that improves the covering of the best grammar. It is performed using pairs of grammars. For example, if the best grammar describes the trajectory of a mobile robot when its goal is to the right, we would expect that another sequence provides information about how to reach a goal to the left of the robot. The generalization process generates a clause that covers both cases calculating the *lgg* (least general generalization [8]) of both clauses. The process can be summarized as follows:

1. Select the best grammar g_{best}
2. Select the grammar g_{other} that provides the largest number of different instantiations of predicates.
3. Compute the *lgg* between grammar rules of g_{best} and g_{other} with different instantiations of predicates and replace the grammar rule from g_{best} by the resulting generalized rule.
4. If the new grammar rule improves the original coverage, it is accepted, otherwise it is discarded.
5. The process continues until a coverage threshold is reached, g_{best} rules cover all the rules in the other grammars or there is no longer improvement with the generalization process.

Table 1. lgg example

c_1	c_2	lgg(c_1,c_2)
pred($State$,$action_2$) ← cond1($State$,$action_1$), cond2($State$,$action_2$).	pred($State$,$action_3$) ← cond1($State$,$action_1$), cond2($State$,$action_3$).	pred($State$,$Action$) ← cond1($State$,$action_1$), cond2($State$,$Action$).

The generalization process is used to produce a more general grammar applicable to different traces of the problem. Table 1 shows the *lgg* of clauses c_1 and c_2 where the constants $action_2$ and $action_3$ are replaced by the variable $Action$.

4 Learning Navigation Tasks

When a mobile robot navigates through an office/house environment it describes a trajectory that can be represented by sequences of actions. Our approach is based on a teleo-reactive framework where learned DCGs represent Teleo-Reactive Programs (TRPs) [9]. TRPs are sets of reactive rules that sense the environment continuously and apply an *action* whose continuous execution eventually satisfies a goal condition. The following basic navigation TRPs are learned in [10]: *wander*, *orient*, *leave-a-trap* and *follow-a-mobile-object*.

Learning to go to a target point. Given a sequence consisting of *wander* and *orient* FOSeq is used to learn a grammar that can go between two places using such skills and possibly inducing intermediate concepts. The user steered the mobile robot with a joystick to reach different goals producing 8 traces. FOSeq learned 8 grammars, one for each trace. After being evaluated, the best induced grammar covered 99.29% of the traces. Table 2 shows the generalized rules learned from the trace of 8 sequences. Predicate names were given by the user. Each rule describes a sub-task along the navigation trajectory. For example, R1 describes the "turning-to-goal" behavior and R2 describes the "direct-to-goal" behavior when the robot does not need to turn because the goal is in its same direction and it wanders to reach the goal. Table 3 shows other hierarchical TRPs learned using other basic skills: *wander*, *orient*, *follow* and *leave-trap*.

Navigation experiments. The experiments were carried out in simulation and with a service ActivMedia robot equipped with a sonar ring and a Laser SICK LMS200 using the Player/Stage software [11]. The goal of the experiments is to show that the learned TRPs can control the robot in completely unknown and dynamic environments. We evaluate the performance of the learned TRPs in 10 different scenarios, with different obstacles' sizes and shapes, and with static and dynamic obstacles. The TRPs were evaluated by the percentage of tasks successfully completed, and the number of operator interventions (e.g., if the robot enters a loop). The robot's initial position and the goal point (when applicable) were randomly generated. In each experiment two operator interventions were allowed, otherwise, the experiment failed. Table 3 summarizes the results

Table 2. Goto TRP rules

(R1) turning-to-goal($State_1$,*go-fwd*)	→ orient($State_1$,*Action*), wander($State_2$,*go-fwd*)
(R2) direct-to-goal($State_1$,*Action*)	→ orient($State_1$,*equal*), wander($State_2$,*Action*)
(R3) cannot-orient($State_1$,*Action*)	→ orient($State_1$,none),wander($State_2$,*Action*)
(R4) ramble(*State*,*Action*)	→ wander(*State*,*Action*)

Table 3. Accuracy: Hierarchical TRPs

TRP	#seq.	Tasks	Int.	Acc1	Acc2
wander + orient (goto)	8	30	2	93.33	86.67
wander + orient + leave-trap	12	30	1	96.67	93.33
follow + wander	10	40	2	95	90
follow + wander + leave-trap	14	40	0	100	100

in simulation. It is shown the number of tasks to test each TRP, the operator interventions and the accuracy with (Acc1) and without interventions (Acc2).

The learned TRPs were integrated as the navigation module of a PeopleBot service robot [12]. The given tasks were: (i) following a human under user commands, (ii) navigating to several places in the environment. Each place has a previously defined name (e.g., kitchen, sofa), (iii) finding one of a set of different objects in a house, and (iv) delivering messages and/or objects between different people. The first three tasks are part of the RoboCup@Home challenge. Navigation and follow videos can be seen at: http://www.youtube.com/user/anon9899

5 Dynamic Gesture Recognition

Interacting with gestures is a natural human ability that can improve the human-computer communication. In this section it is described how to learn grammars from sequences of dynamic gestures and use them to classify new sequences.

FOSeq transforms low-level information from sensors into a relational representation. We have sequences of state-value pairs, where a value can be an action or a boolean variable, as described below. We used a database[1] of 7308 samples from 9 dynamic gestures taken from 10 men and 5 women. Figure 2(a) shows the initial and final position for each gesture. The whole set can be seen in Figures 2(b)-(j). Gestures were executed with the right arm and they were obtained using the monocular visual system described in [6].

Each sequence is a vector with sets of seven attributes describing the executed gesture. An example of a sequence is: $(+ + -- \text{T F F}),(+ + - + \text{T F F}), \ldots$ where the first three attributes of each vector describe motion and the rest describe posture. Motion features are $\Delta area$, Δx and Δy, representing changes in hand area and changes in hand position of the XY-axis of the image respectively. These features take one of three possible values: $\{+,-,0\}$ indicating increment,

[1] Database available at http://sourceforge.net/projects/visualgestures/

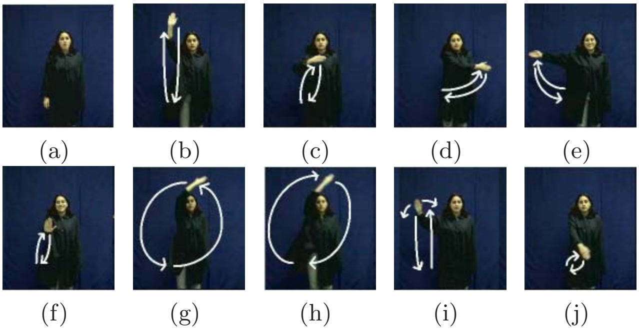


Fig. 2. Gesture set: (a) initial and final position, (b) attention, (c) come, (d) left, (e) right, (f) stop, (g) turn-right, (h) turn-left, (i) waving, (j) pointing

decrement or no change, according to the area and position of the hand in a previous image. Posture features are: *form, above, right* and *torso*, and describe hand appearance and spatial relations between the hand and face/torso. Hand appearance is described by *form*. Possible values for this feature are {+,−,0}: (+) if the hand is vertical, (−) if the hand has horizontal position, and (0) if the hand is tilted to the right or left over the XY plane. Features *right* and *torso* indicate if the hand is to the right of the head and over the torso. Based on this information, sequences are transformed into a first-order representation by replacing their attributes with a meaningful fact (e.g., hmove($State$, $right$), vmove($State$, up),size($State$,inc), shape($State$, $vertical$), ...).

We focused on the recognition between gestures produced by one person following the experimentation setting described in [6]: (i) from 50 sequences of each gesture, randomly select 20 sequences to learn the grammars and build training sets of 2 and 10 sequences, (ii) learn a grammar for each gesture, (iii) test the grammars with the remaining 30 sequences, (iv) repeat 10 times.

The overall accuracy obtained by FOSeq and the HMM approach in [6] is as follows: both training sub-sets FOSeq performs similar to HMMs: with 2 training sequences, FOSeq got 95.17%, whereas HMMs 94.85%. With 10 training sequences, FOSeq got 97.34%, and HMMs 97.56%. These results are very promising as HMM is the leading technique in this application. Table 4 shows the confusion matrix that expresses the proportion of true classified instances for 10 training sequences. The best classified gestures are: *left, right, turn-right* and *waving* (100%) whereas *pointing* is the worst classified gesture (85.17%). Misclassification are concentrated between *pointing* and *come*.

Learned grammars for *pointing* and *come* have 5 common rules whereas *right* and *left* grammars do not have any. For instance, an identical grammar rule for *pointing*/*come* is: R1 → above_face($State$,$false$) over_torso($State$,$true$) explaining that the hand is not near the face but it is over the torso. This type of similarities and the identification of common sub–gestures is not possible to obtain with other approaches. Learning relational grammars for gesture recognition produces an explicit representation of rules and is able to identify and

Table 4. Confusion matrix for 10 sequences. Classes: 1) *attention*, 2) *come*, 3) *left*, 4) *pointing*, 5) *right*, 6) *stop*, 7) *turn-left*, 8) *turn-right*, 9) *waving*

	1	2	3	4	5	6	7	8	9	
1	**99.33**							0.67		300
2		**97.24**		2.76						290
3			**100**							290
4		13.10	1.72	**85.17**						290
5					**100**					300
6	1.03					**97.59**			1.38	290
7					2.07		**96.55**		1.38	290
8								**100**		290
9									**100**	290
	301	320	295	255	306	283	280	292	298	2630

generate rules for sub-gestures. It also helps to find similarities between different gestures and has a competitive performance against HMMs.

6 Conclusions and Future Work

In this work we have introduced an algorithm called FOSeq, that takes sequences of states and actions and induces a grammar able to parse and reproduce the sequences. FOSeq learns a grammar for each sequence, followed by a generalization process between the best evaluated grammar and other grammars to produce a generalized grammar covering most of the sequences. Once a grammar is learned it is transformed into a TRP in order to execute particular actions and achieve a goal. FOSeq was able to learn a navigation grammar from sequences given by the user and used the corresponding TRP to guide the robot in several navigation tasks in dynamic and unknown environments. FOSeq was also used to learn grammars from gesture sequences with very competitive results when compared with a recent state-of-the-art system. As part of our future work, we are working on learning more TRPs to solve other robotic tasks, we plan to extend the experiments with gestures to obtain a general grammar for a gesture performed by more than one person, and we are interested in reproducing gestures with a manipulator.

The authors thank the reviewers for their useful comments. This research was sponsored by CONACYT under grant 203873 .

References

1. Adriaans, P.W., Trautwein, M., Vervoort, M.: Towards high speed grammar induction on large text corpora. In: Jeffery, K., Hlaváč, V., Wiedermann, J. (eds.) SOFSEM 2000. LNCS, vol. 1963, pp. 173–186. Springer, Heidelberg (2000)
2. van Zaanen, M.V.: Abl: alignment-based learning. In: Proc. 17th Conf. on Computational linguistics, Association for Computational Linguistics, pp. 961–967 (2000)

3. Bernard, M., de la Higuera, C.: Gift: Grammatical inference for terms. In: International Conference on Inductive Logic Programming (1999)
4. Nevill-Manning, C., Witten, I.: Identifying hierarchical structure in sequences: A linear-time algorithm. Journal of Artificial Intelligence Research 7, 67–82 (1997)
5. Amit, R., Mataric, M.J.: Learning movement sequences from demonstration. In: ICDL 2002: Proc. 2nd International Conf. on Development and Learning, Cambridge, MA, pp. 12–15 (2002)
6. Avilés-Arriaga, H., Sucar, L., Mendoza, C.: Visual recognition of similar gestures. In: 18 International Conference on Pattern Recognition ICPR 2006, vol. 1, pp. 1100–1103 (2006)
7. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. 20th Int. Conf. Very Large Data Bases, VLDB, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
8. Plotkin, G.: A note on inductive generalization. Machine Intelligence 5, 153–163
9. Benson, S., Nilsson, N.J.: Reacting, planning, and learning in an autonomous agent. Machine Intelligence 14, 29–62 (1995)
10. Vargas, B., Morales, E.F.: Learning navigation teleo-reactive programs using behavioural cloning. In: IEEE International Conference on Mechatronics (2009)
11. Vaughan, R.T., Gerkey, B.P., Howard, A.: On device abstractions for portable, reusable robot code. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2421–2427 (2003)
12. Avilés, H.H., Sucar, L.E., Morales, E.F., Vargas, B.A., Sánchez, J., Corona, E.: Markovito: A flexible and general service robot, January 2009. Studies in Computational Intelligence, vol. 177, pp. 401–423. Springer, Heidelberg (2009)

On Environmental Model-Based Visual Perception for Humanoids

D. Gonzalez-Aguirre, S. Wieland, T. Asfour, and R. Dillmann

Institute for Anthropomatics, University of Karlsruhe,
Haid-und-Neu-Strasse 7, Karlsruhe-Germany
{gonzalez,wieland,asfour,dillmann}@ira.uka.de

Abstract. In this article an autonomous visual perception framework for humanoids is presented. This model-based framework exploits the available knowledge and the context acquired during global localization in order to overcome the limitations of pure data-driven approaches. The reasoning for perception and the *properceptive*[1] components are the key elements to solve complex visual assertion queries with a proficient performance. Experimental evaluation with the humanoid robot ARMAR-IIIa is presented.

Keywords: Model-Based Vision, Object Recognition, Humanoids.

1 Introduction

The emerging research field of humanoid robots for human daily environment is an exciting multidisciplinary challenge. In order to properly and effectively interact and operate within those environments it is indispensable to equip the humanoid robots with autonomous perception capabilities.

Recently, considerable results in this field have been achieved (see [1],[2]) and several humanoid robots exposed various knowledge-driven capabilities and skills. However, those approaches mainly concentrate on knowledge processing for graspable objects with fixed object-centered attention zones, e.g. kettle tip while pouring tea or water faucet while washing a cup.

These approaches assume a fixed pose of the robot in its environment in order to perceive and manipulate objects and environmental elements within a kitchen. In addition, the very narrow field of view with no objects in the background constrains their applicability in real daily scenarios.

These perception limitations can be overcome through an enhanced exploitation of the available model and knowledge information by including a reasoning sublayer within the visual perception system. There exist works on humanoids reasoning for task planning and situation interpretation, see [3], [4]. However, they focus on atomic operations and discrete transitions between states of the

[1] Capturing the world through internal means, e.g. models and knowledge mechanisms. It is the counterpart of perception which captures the world through external means, sensory stimuli.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 901–909, 2009.

modeled world for behavior generation and verification. This high-level reasoning is not the focus of the present work, but the inclusion of the essential reasoning mechanism while perception takes place in order to robustly recognize and interpret complex patterns, i.e. distinguish and track environmental objects in presence of cluttered backgrounds, grasping occlusions and different poses of both, the humanoid and the environmental objects.

The processing of low-level sensor data and higher-level knowledge model for segmentation, rejection and recognition constitutes the reasoning for visual perception. It bridges the gap between the image processing and object recognition components through a cognitive perception framework [5].

In order to make this reasoning mechanism tractable and its implementation plausible it is necessary to profit from both the *model-to-vision* coupling resulting from the model-based approach and the *vision-to-model* association acquired during the global localization by means of our previous work, see [6], [7].

The focus herein is placed on rigid elements of the environment which could be transformed through rigid-parametric transformations, e.g. furniture, kitchen appliances, etc.

In the following sections the visual perception framework is introduced along experimental results of the demonstration application scenario where these concepts were implemented and evaluated providing remarkable real-time results which pure data driven algorithms would hardly provide.

2 Visual Perception Framework

The visual perception framework extracts valuable information from the real world through stereo color images and kinematic configuration of the humanoids active vision head. The adequate representation, unified and efficient storage, automatic recall and task-driven processing of this information take place within different layers (so called *states of cognition*) of the framework.

Latter cognition states are categorically organized according to [5] as *sensing*, *attention*, *reasoning*, *recognition*, *planning*, *coordination* and *learning*. In this manner three *principal cycles* arise, namely *perception-cycle*, *coordination-cycle* and *learning-cycle*, see Fig.1.

Memory; World Model and Ego Spaces. The formal representation of real objects within the application domain and the relationships between them constitutes the *long term memory*, i.e. the world-model. In our approach appropriate description has been done by simultaneously separating the geometric composition from the pose and encapsulating the attributes which correspond to the configuration of the instances, e.g. name, identifier, type, size, parametric transformation, etc. This structure, along the implemented mechanism for graph pruning and inexact matching lay down the *spatial query solver* used in Sec.4.

On the other hand, the *mental imagery* (see Sec.3.1) and the acquired percepts are contained within an ego centered space which corresponds to the *short term memory*.

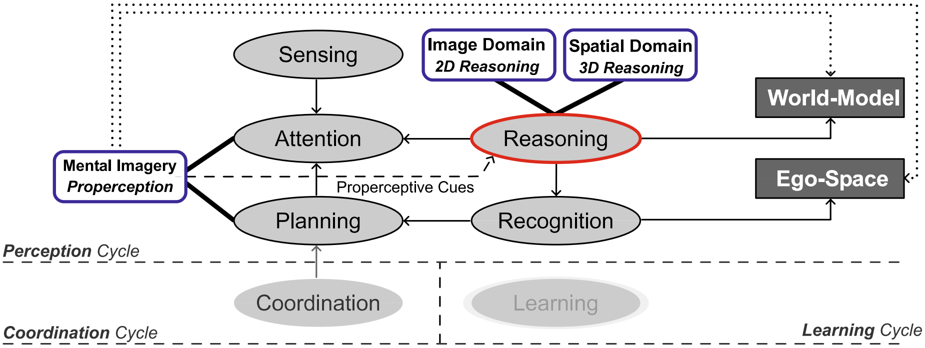


Fig. 1. The perception framework, including states of cognition and principal cycles

3 Visual Sensing and Planning

Sensing. The noise tolerant vision-to-model coupling arise from the full configuration of the active vision system including the internal joint configuration, external position and orientation of the cameras centers as well as all required mechanisms to obtain euclidean metric from stereo images, see [8], [9].

Planning. It involves three fundamental aspects;

First, once the visual target-node has been established, it provides a frame and the definition of a subspace Ψ where the robot has to be located, therewith the target-node can be robustly recognized, see Fig.2-a,b. Note that this subspace Ψ is not a single pose as in [3] and [4], but a wide range of reachable poses allowing more flexible applicability and more robustness through wider tolerance for uncertainties in the navigation and self-localization.

Subsequently, the visual-planner uses the restricted subspace and target node frame to generate a transformation from the current pose to a set of valid poses. These poses are used as input of the navigation layer [10] to be unfolded and executed.

Finally, once the robot has reached the desired position, the visual-planner uses the description of the node to predict parametric transformations and appearance properties, namely, how the image content should look like, and how the spatial distribution of environmental elements is related to the current pose. Note that this is not a set of stored image-node associations (as in the appearance graph) but a novel generative-extractive continuous approach implemented by the following properception mechanism.

3.1 Mental Imagery

The properception skills *prediction* and *cue extraction* allow the humanoid to capture the world through internal means by synergistically exploiting both, the world-model (full scene-graph) and the *hybrid virtual cameras*, see Fig.3. These virtual devices use the plenary stereoscopic calibration of the real stereo rig in

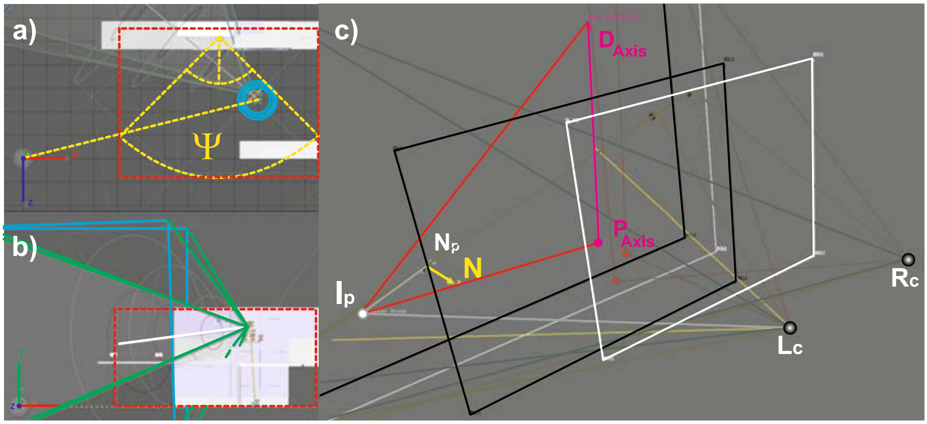


Fig. 2. a) Example of restriction subspace Ψ where the target node can be robustly recognized, top view of the kitchen. b) Ψ side view. c) Geometric elements involved during the spatial reasoning for perception.

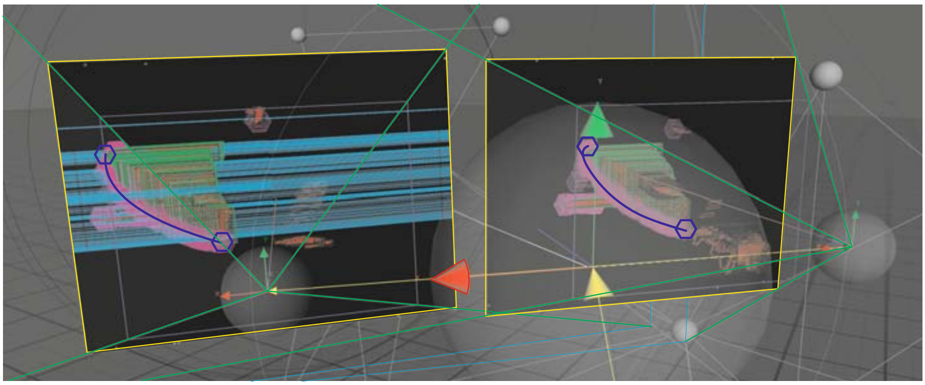


Fig. 3. The properceptive mental imagery for trajectory prediction. Note that the blue lines in the left and right image planes of the hybrid virtual cameras show the ideal trajectory of the interest point (door handle end-point) during the door opening. This predicted subspace reduce region of interest and helps to reject complex outliers, see example in Fig.4.

order to set the projection volume and matrix within the virtual visualization, a common practice in the augmented reality [11] for overlay image composition. However, this hybrid virtual stereo rig is used to predict and analyze the image content within the world-model, including the previously described parametric transformations, extraction of position and orientations cues either for static or dynamic configurations.

4 Visual Reasoning for Recognition

The reasoning process for perception is decomposed in two domains; the visual domain (2D reasoning) which concerns with the image content and the spatial domain (3D reasoning), which manages the geometric content.

Visual Domain. The pose estimation of a partial occluded door handle, when the robot has already grasped it, turns out to be difficult because of many perturbation factors. No size rejection criteria may be assumed, because the robot hand is partially occluding the handle surface and the hand slides during task execution, producing variation of the apparent size. No assumption about the background of the handle could be made, because when the door is partially open and the perspective view overlaps handles from lower doors similar chromatic distribution appear. In addition, the glittering of the metal surfaces on both, the robots hand and doors handle, produce very confusing phenomena, when using standard segmentation techniques [12].

In this context, we propose an environment dependent but very robust and fast technique (15-20 ms) to simultaneously segment the regions and erode the borders, producing non-connected regions which suits our desired preprocessing-filtering phase. First, the raw *RGB*-color image $I_{rgb}(x, y) \in \mathbb{N}^3$ is split per channel and used to compute the *power image* I_ϕ, see Fig.4

$$I_\phi(x, y, n) = [I_r(x, y) \cdot I_g(x, y) \cdot I_b(x, y)]^n, \text{ where } n \text{ and } I_\phi(x, y, n) \in \mathbb{R}.$$

After linear normalization and adaptive thresholding, a binary image $I_B(x, y)$ is produced, which is used to extract the blobs B_k and build feature vectors for rejection purposes. The feature vector $F(B_k)$ is formed by the blobs area $\omega(B_k)$, the energy density $\delta(B_k)$, and the elongation descriptor, i.e. the ratio of the eigen values $E_{\sigma_i}(B_k)$ of the energy covariance matrix[2] expressed by

$$F(B_k) := [\delta(B_k), \omega(B_k), E_{\sigma_1}(B_k)/E_{\sigma_2}(B_k)].$$

This characterization enables a powerful rejection of blobs when verifying the right-left cross matching by only allowing candidates in pairs (B_k, B_m) where the criterion is fulfilled, i.e. the orientation of their axis shows a discrepancy less than arccos(K_{min}) radians, i.e.

$$K(B_k, B_m) := \|E_{\sigma_1}(B_k) \cdot E_{\sigma_1}(B_m)\| > K_{min}.$$

The interest point I_p in both images is selected as the furthest pixel along the blobs main axis in opposed direction of the vector Γ_R, i.e. unitary vector from the door center to the center of the line segment where the rotation axis is located, see Fig.2-c. This vector is obtained from the mental imagery as stated in Sec.3.1-c. Moreover, the projected edges of a door within the kitchen aid the segmentation phase to extract the door pose and improves the overall precision by avoiding to consider edges pixels close to the handle.

[2] From the power image by selecting masked blobs $\omega(B_k)$ in the binary image $I_B(x, y)$.

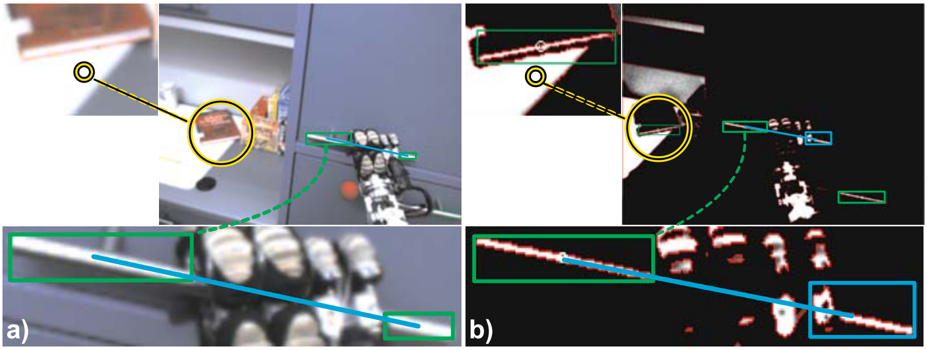


Fig. 4. a) Input image I_{rgb}. Note that the book (particularly the white paper side) in the background shows not only similar color distribution, but almost the same size of the door handle. b) The power image I_{ϕ}. Based only on the pure data-driven classification it will be hardly possible to reject the presence of a handle within the location of the book.

The key factor of this model-to-vision coupling relies on the fact that very general information is applied, i.e. from the projected lines and blobs employing mental imagery, only their direction is used (e.g. noise-tolerant criterion K_{min}) and not the position itself, which differs from the real one, due to the discretization, quantization, noise and uncertainties.

Spatial Domain. One of the most interesting features of our approach is the usage of the vision-to-model coupling to deal with limited visibility.

In order to provide the required information from the global planner or coordinator module it is necessary to estimate the interest point I_p, and the normal vector N_p of the grasping element, see Fig.2-c, e.g. the door handle. Because of the size of both, the door and the 3D field of view (3DFOV, see Fig.2-a,b), it can be easily corroborated that the minimal distance, where the robot must be located for the complete door to be contained inside the robots 3DFOV, lies outside of the reachable space, therefore common triangulation techniques may not be used. In this situation, the module reasoning for perception switches from pure data driven algorithm to the following recognition method which only requires three partially visible edges of the door and uses the context (robot pose) and model to assert the orientation of the door's normal vector and the door's angle of aperture.

First, a 2D-line $\varUpsilon_i$ on an image and the center of its capturing camera L_c or R_c define a 3D-space plane $\varPhi_{(i,j)}$, see Fig.2-c. Hence, two such planes $\varPhi_{(L,L)}$ and $\varPhi_{(\mu(\varUpsilon_L,\varUpsilon_R),R)}$, resulting from the matching $\mu(\varUpsilon_L,\varUpsilon_R)$ of two lines in the left and right images in a stereo system define an intersection subspace, i.e. a 3D-line

$$\varLambda_i = \varPhi_{(L,L)} \wedge \varPhi_{(\mu(\varUpsilon_L,\varUpsilon_R),R)}.$$

These 3D-lines Λ_i are subject to noise and calibration artifacts. Thus, they are not suitable to compute 3D intersections. However, their direction is robust enough.

Next, the left image 2D points $H_{(L,i)}$ resulting from the intersection of 2D-lines Υ_i are matched against those in the right image $H_{(R,j)}$ producing 3D points $X_{(R,j)}$ by means of triangulation in a minimal-square fashion.

Finally, it is possible to acquire corners of the door and directions of the lines connecting them, even when only partial edges are visible. Herein, the direction of the vector Γ_R is the long-term memory cue used to select the 3D line edge direction D_{Axis} and its point P_{Axis}.

5 Experimental Evaluation

In order to demonstrate the advantages of the presented framework for visual perception and to verify our methods, we accomplished the task of door opening in a regular kitchen with the humanoid robot ARMAR-IIIa [10], see Fig.5.

In this scenario, the estimation of the normal vector N_p, and therewith the minimization of the external forces on the hand, because the door changes its orientation during manipulation. In our previous approach [13] the results using only one sensory channel (force-torque sensor) are acceptable but not satisfactory, because the estimation of the task frame depends on the accuracy of the robot kinematics.

In this experimental evaulation the framework estimates the interest point and normal vector of the door, therewith the task frame. During task execution this frame is estimated by the before mentioned methods and the impedance control is balancing the external forces and torques at the hand, caused by vision artifacts. Robustness and reliability of the handle tracker are the key to reduce the force stress in the robots wrist as it can easily be seen in Fig.6.

Combining stereo vision and force control provides the advantage of real-time task frame estimation by vision, which avoids the errors of the robots kinematics and adjustment of actions by the force control.

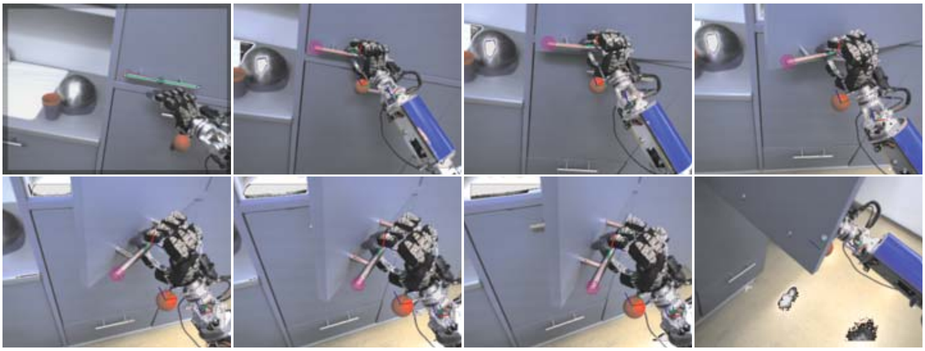


Fig. 5. Experimental evaluation of the perception framework

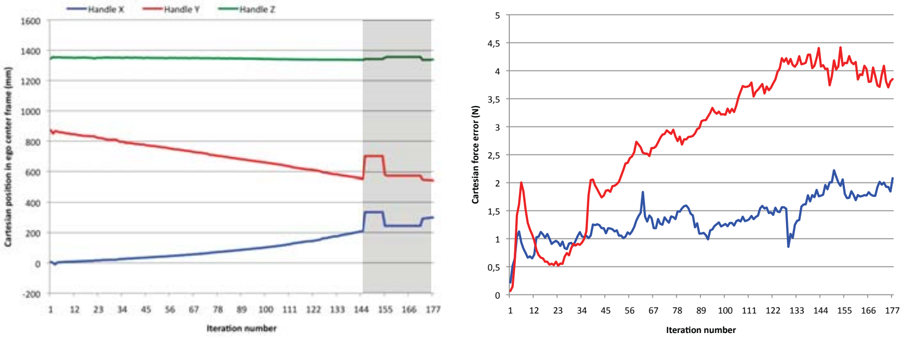


Fig. 6. Left; Cartesian position of the handle midpoint. Smooth movement in the three cartesian dimensions, until iteration 144 when the handle is completely occluded. **Right**; Total stress forces at the task frame. The red plot represents the force in the pulling direction using only force-torque sensor and previous kinematic configuration. The blue plot is the result when applying a vision-based estimation of the task frame in a sensor fusion fashion.

6 Conclusions

The world-model and the available context acquired during self-localization will not only make it possible to solve, otherwise hardly possible, complex visual assertion queries, but it will also dispatch them with a proficient performance. This is possible through the introduced perception framework which implements the basic reasoning skills by extracting simple but compelling geometrical cues from the properception component and then applying them as filters for the classification of percepts, tracking and optimization of the region of interest (in terms of size and speed) and handling of incomplete visual information, see Fig. 5. The coupling of model-to-vision by means of the properceptive cues generated by the mental imagery along with the visual and spatial reasoning mechanism for perception are the main novel contributions of the framework. A more general formulation, exploitation and exploration of these ideas are the main axis of our current work.

Acknowledgments

The work described in this article was partially conducted within the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft) and the EU Cognitive Systems projects GRASP (FP7-215821) and PACO-PLUS (FP6-027657) funded by the European Commission.

References

1. Okada, K., Kojima, M., Sagawa, Y., Ichino, T., Sato, K., Inaba, M.: Vision based behavior verification system of humanoid robot for daily environment tasks. In: IEEE-RAS Int. Conference on Humanoid Robots (2006)
2. Okada, K., Kojima, M., Tokutsu, S., Maki, T., Mori, Y., Inaba, M.: Multi-cue 3D object recognition in knowledge-based vision-guided humanoid robot system. In: IEEE/RSJ International Conference on Intelligent Robots and Systems 2007 (2007)
3. Okada, K., Tokutsu, S., Ogura, T., Kojima, M., Mori, Y., Maki, T., Inaba, M.: Scenario controller for daily assistive humanoid using visual verification, task planning and situation reasoning. Intelligent Autonomous Systems 10 (2008) ISBN 978-1-58603-887-8
4. Okada, K., Kojima, M., Tokutsu, S., Mori, Y., Maki, T., Inaba, M.: Task guided attention control and visual verification in tea serving by the daily assistive humanoid HRP2JSK. In: IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems (2008)
5. Patnaik, S.: Robot Cognition and Navigation: An Experiment with Mobile Robots. Springer, Heidelberg (2007)
6. Gonzalez-Aguirre, D., Asfour, T., Bayro-Corrochano, E., Dillmann, R.: Model-based visual self-localization using geometry and graphs. In: 19th International Conference on Pattern Recognition. ICPR 2008 (2008)
7. Gonzalez-Aguirre, D., Asfour, T., Bayro-Corrochano, E., Dillmann, R.: Improving Model-Based Visual Self-Localization using Gaussian Spheres. In: 3rd International Conference on Applications of Geometric Algebras in Computer Science and Engineering. AGACSE 2008 (2008)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, San Francisco (2004)
9. The Integrating Vision Toolkit (IVT), http://ivt.sourceforge.net/
10. Asfour, T., Regenstein, K., Azad, P., Schroder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R.: ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In: IEEE-RAS Int. Conference on Humanoid Robots (2006)
11. Gordon, G., Billinghurst, M., Bell, M., Woodfill, J., Kowalik, B., Erendi, A., Tilander, J.: The use of dense stereo range data in augmented reality. In: Proceedings of International Symposium on Mixed and Augmented Reality, 2002. ISMAR 2002, pp. 14–23 (2002)
12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
13. Prats, M., Wieland, S., Asfour, T., del Pobil, A.P., Dillmann, R.: Compliant interaction in household environments by the Armar-III humanoid robot. In: IEEE-RAS Int. Conference on Humanoid Robots (2008)

Dexterous Cooperative Manipulation with Redundant Robot Arms

David Navarro-Alarcon, Vicente Parra-Vega,
Silvionel Vite-Medecigo, and Ernesto Olguin-Diaz

Grupo de Robótica y Manufactura Avanzada, CINVESTAV-Saltillo, México
{david.navarro,vicente.parra,silvionel.vite}@cinvestav.edu.mx

Abstract. A novel model-based force-position control scheme for cooperative manipulation tasks with redundant arms is proposed in this paper. Employing an orthogonal decomposition of the object contact mechanics, independent pose and force trajectory tracking can be achieved. In this way, a high precision cooperative scheme is enforced since the projection of the object velocity into the contact normal direction converges to zero, improving the system cooperativeness. Simulation results are presented for a humanoid torso to visualize its closed-loop performance.

1 Introduction

Multi-arm cooperative schemes have been a topic of special interest for the last 20 years. This comes from the fact that multiple arms working in a given task can improve the execution dexterity, increase the pay load capability, provide higher manipulation flexibility, among other advantages [1]. Moreover, multi-arm cooperative schemes can be employed to model certain robot manipulation tasks such the robotic hand wherein each finger is modeled as an arm, and the humanoid robot manipulation where two anthropomorphic arms manipulate an object. Notice that these manipulation tasks involve physical interaction among the cooperators, consequently, contact forces arise when manipulating an object. Then, in order to provide a stable physical interaction, the object position trajectory and the exerted contact force must be simultaneously controlled among the participants. It must be remarked that these two problems are not fully decoupled as the case of fixed holonomic constraints [2]. Therefore, in cooperative schemes the object is now allowed to move in the same direction as the exerted contact force.

On the other hand, when redundant robot arms, like the humanoid anthropomorphic arms, are employed for cooperative tasks the benefits and possibilities are increased even more. This as a result of the multiple kinematic configurations available for a given end-effector pose (position+orientation). Even more, with the use of redundancy a cooperative system can avoid collisions with obstacles while manipulating the object, reconfigure itself in order to optimize the power consumption while exerting a given contact force, or just to optimize a meaningful cost function. In this sense, redundancy can help to improve the dexterity of the robotic system.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 910–917, 2009.

The main contribution of this paper is the enforced stability in the Lyapunov sense of the normal velocity-position error manifold, presented here as *cooperativeness error*. Therefore, the stable execution of independent force and position tracking is guaranteed in the normal direction at the contact point. This control scheme is useful for humanoid cooperative manipulation, where is required a stable physical interaction among arms. To present this result, Section 2 introduces the nature of the dynamical problem of cooperative manipulation with redundant arms, and the proposed parametrization of the open-loop error dynamics. Section 3 describes the design procedure for the passivity-based control law. Simulation results with a humanoid robot with two redundant arms are presented in Section 4, with final discussions in Section 5.

2 Mathematical Modeling

2.1 Robot Kinematics

Consider a robotic system composed of two redundant arms with similar kinematic structure. The joint position coordinate for each redundant arm is given by $q_1, q_2 \in \Re^n$. In this work, we will assume that both redundant arms have the same number of degrees of freedom (DoF). Then, the forward and differential kinematics equations of each arm are given by

$$X_i = f_(q_i), \quad \dot{X}_i = J_i(q_i)\dot{q}_i \tag{1}$$

with $X_i \in \Re^m$ as the end-effector pose and $J_i(q_i) = \frac{\partial f_i(q_i)}{\partial q_i} \in \Re^{m\times n}$ as the Jacobian matrix, for $i = 1, 2$. Since both arms are redundant there is not unique solution for the inverse kinematics problem because $n > m$, consequently the non-square Jacobian matrix can not be inverted. This apparent complexity turns into an attribute if we encode two different tasks within a single joint velocity desired vector $\dot{q}_{di} \in \Re^n$[1]. The former is the usual tracking task of the end-effector while the latter may be a reference to reconfigure the kinematic chain [3]:

$$\dot{q}_{di} = \dot{q}_{Pi} + \dot{q}_{Ki} \tag{2}$$

Then the challenge is to design $\dot{q}_{Pi}$ and $\dot{q}_{Ki}$ to encode the end-effector desired velocity $\dot{X}_{di}$ and the kinematic reconfiguration to satisfy a given cost function, respectively [2]. A common approach is to use the Jacobian right pseudo-inverse $J_i^+(q_i) = J_i^T(q_i)\left[J_i(q_i)J_i^T(q_i)\right]^{-1} \in \Re^{n\times m}$ and its orthogonal projection matrix $I - J_i^+(q_i)J_i(q_i) \in \Re^{n\times n}$ to span the Jacobian kernel. Then, we define $\dot{q}_{Pi} = J_i^+(q_i)\dot{X}_{d_i}$ and $\dot{q}_{Ki} = [I - J_i^+(q_i)J_i(q_i)]\frac{\partial \Omega_i(q_i)}{\partial q_i}$, where $\Omega_i(q_i) \in \Re$ is a cost function to be locally optimized. The desired joint position can be computed by integrating the desired velocity $q_{di} = \int_{t_0}^{t} \dot{q}_{di}\, \mathrm{d}t + q_i(t_0)$.

[1] Along this work, subindex d and r are used to denote the desired and reference values of a given variable, respectively.

[2] Notice that these vectors satisfy the following $\dot{q}_{Pi}^T\dot{q}_{Ki} = 0$, *i.e.* they are orthogonal. Therefore, $\dot{q}_i \rightarrow \dot{q}_{di}$ implies the achievement of both $\dot{q}_{Pi}$ and $\dot{q}_{Ki}$.

2.2 Holonomic Cooperation

In order to independently control the object motion and the interaction forces, a meaningful mathematical description of the mobile holonomic constraint (object) must be synthesized. To this end, consider the following expression of the holonomic constraint $\varphi(X_1, X_2) \in \Re$ imposed by the object over the cooperative system

$$\varphi(X_1, X_2) = \varphi_1(X_1) + \varphi_2(X_2) = 0 \tag{3}$$

Notice that (3) means that the robotic system is in contact with a rigid object. Then, both arms must satisfy (3) when performing the trajectory. It is clear that the independent forward kinematics of each arm X_1 and X_2 can be arranged such that the scalar equation representing the holonomic constraint satisfies $\varphi_1(X_1) + \varphi_2(X_2) = 0$, where $\varphi_1(X_1) \in \Re$ and $\varphi_2(X_2) \in \Re$ are scalar functions of each arm, dependant only in the corresponding end-effector's pose.

Our controller is based on the orthogonal decomposition of the contact mechanics between the object and the arms [2]. To this end, consider the following definition of the normal subspace span horizontal vector $J_{\varphi i}(q_i) \in \Re^{1\times n}$:

$$J_{\varphi i}(q_i) = \frac{\nabla\varphi_i(X_i)}{\|\nabla\varphi_i(X_i)\|} J_i(q_i) \tag{4}$$

where $\nabla\varphi_i(X_i) = \frac{\partial\varphi_i(X_i)}{\partial X_i} \in \Re^{1\times m}$ stands for the gradient of $\varphi_i(X_i)$. The term $\frac{\nabla\varphi_i(X_i)}{\|\nabla\varphi_i(X_i)\|}$ is a unit operational vector which points out at the contact normal direction. Consider the following definition of the orthogonal projection matrix $Q_{\varphi i}(q_i) \in \Re^{n\times n}$ which spans the tangent subspace at the contact point:

$$Q_{\varphi i}(q_i) = I - J^{+}_{\varphi i}(q_i) J_{\varphi i}(q_i) \tag{5}$$

where $I \in \Re^{n\times n}$ is an identity matrix and $J^{+}_{\varphi i}(q_i) = J^{T}_{\varphi i}(q_i)\left[J_{\varphi i}(q_i)J^{T}_{\varphi i}(q_i)\right]^{-1} = \frac{J^{T}_{\varphi i}(q_i)}{\|J_{\varphi i}(q_i)\|^2} \in \Re^{n}$ denotes the right pseudo-inverse of $J_{\varphi i}(q_i)$ which always exists

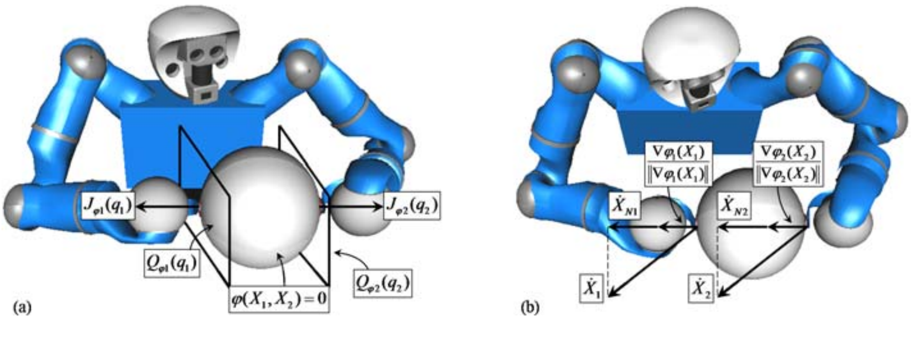


Fig. 1. (a) The vector $J_{\varphi i}(q_i)$ and the matrix $Q_{\varphi i}(q_i)$ span the normal and tangent subspaces at the contact point, respectively. (b) Conceptual representation of the end-effector constrained velocity $\dot{X}_{Ni}$, which points onto the normal direction given by the unit vector $\frac{\nabla\varphi_i(X_i)}{\|\nabla\varphi_i(X_i)\|}$.

since $J_{\varphi i}(q_i)J_{\varphi i}^T(q_i) = \|J_{\varphi i}(q_i)\|^2 \neq 0, \ \forall q_i \in \Re^n$. It is evident that $Q_{\varphi i}(q_i) + J_{\varphi i}^+(q_i)J_{\varphi i}(q_i) = I$, this way the normal and tangent projections of $\dot{q}_i$ are:

$$\dot{q}_i = Q_{\varphi i}(q_i)\dot{q}_i + J_{\varphi i}^+(q_i)\dot{X}_{Ni} \tag{6}$$

where the scalar $\dot{X}_{Ni} = J_{\varphi i}\dot{q}_i = \frac{\nabla\varphi_i(X_i)}{\|\nabla\varphi_i(X_i)\|}\dot{X}_i \in \Re$ stands for the constrained velocity [4] which is the normal component of the total operational velocity $\dot{X}_i$ (see Fig. 1b). Notice that both arms satisfy $\dot{X}_{N1} + \dot{X}_{N2} = 0$ when manipulating the object. This result can be easily proved by taking the time derivative of (3) as follows $\frac{d}{dt}\varphi(X_1, X_2) = \nabla\varphi_1(X_1)\dot{X}_1 + \nabla\varphi_2(X_2)\dot{X}_2 = 0 \rightarrow \dot{X}_{N1} + \dot{X}_{N2} = 0$.

2.3 Cooperativeness Error

It can be said that the cooperative arms form a closed kinematic chain which exhibits a passive joint (i.e. a non actuated joint) at the contact point. Then, a certain degree of coordination must be enforced to ensure the object manipulation while exerting a given contact force, both at the same normal direction, exactly at the two both passive joints. Notice that the traditional hybrid force-position control [5] for holonomic constraints independently controls the force and position errors by projecting them into a normal and tangent subspace, respectively. However, this is valid only when the constraint is fixed not when the constraint is moving, such as an object being manipulated, because of the principle of virtual work. Therefore, in this paper we introduce the notion of cooperativeness error $S_{xi} = \dot{X}_{Ni} - \dot{X}_{Nri} \in \Re$ that arises as a velocity-position error manifold mapped into the normal direction and whose convergence denotes the control of the normal trajectories. The constrained velocity reference value can be computed by:

$$\dot{X}_{Nri} = \frac{\nabla\varphi_i(X_i)}{\|\nabla\varphi_i(X_i)\|}[\dot{X}_{di} - (X_i - X_{di})] \tag{7}$$

where $X_{di}, \dot{X}_{di} \in \Re^m$ are the desired end-effector trajectory. The cooperativeness error S_{xi} also give us an indirect measure of the undesirable pushing-pulling effects among the robot arms, thus, the convergence of S_{xi} will enforce a stable robot physical interaction.

2.4 Robot Constrained Dynamics

Consider a rigid and fully actuated robotic torso composed of two redundant arms[3] manipulating a rigid object. The dynamic equation is given then by the canonical Euler-Lagrange formulation as follows:

$$H_1(q_1)\ddot{q}_1 + [C_1(q_1, \dot{q}_1) + B_1]\,\dot{q}_1 + g_1(q_1) = \tau_1 + J_{\varphi 1}^T(q_1)\lambda \tag{8}$$

$$H_2(q_2)\ddot{q}_2 + [C_2(q_2, \dot{q}_2) + B_2]\,\dot{q}_2 + g_2(q_2) = \tau_2 + J_{\varphi 2}^T(q_2)\lambda \tag{9}$$

[3] It is assumed that no dynamic/kinematic coupling exists among both arms, *i.e.*, each arm is represented by an independent dynamic model. Coupling will arise through the interaction forces.

where for the i-th arm, $H_i(q_i) \in R^{n\times n}$ denotes the inertia matrix, $C_i(q_i,\dot{q}_i) \in \Re^{n\times n}$ denotes the Coriolis matrix, $B_i \in \Re^{n\times n}$ is the damping matrix, $g_i(q_i) \in \Re^n$ represents the gravity loads, and $\tau_i \in \Re^n$ stands for the joint input torques. The scalar $\lambda \in \Re$ represents the magnitude of the operational force vector $F_i \in \Re^m$ exerted from the manipulated object to robot arm. Physically, the product $J_{\varphi i}^T(q_i)\lambda = J_i^T(q_i)\frac{\nabla\varphi_i^T(X_i)}{\|\nabla\varphi_i(X_i)\|}\lambda$ stands for the joint torque distribution onto the arm from the manipulated object exerted force. Notice that λ is related with F_i as follows $F_i = \frac{\nabla\varphi_i^T(X_i)}{\|\nabla\varphi_i(X_i)\|}\lambda$.

Now, the left hand side of (8)-(9) can be linearly parameterized in terms of a joint nominal reference[4] $\dot{q}_{ri} \in \Re^n$ as follows [6]:

$$H_i(q_i)\ddot{q}_{ri} + [C_i(q_i,\dot{q}_i) + B_i]\dot{q}_{ri} + g_i(q_i) = Y_{ri}\Theta_i \tag{10}$$

where the regressor $Y_{ri} = Y_{ri}(q_i,\dot{q}_i,\dot{q}_{ri},\ddot{q}_{ri}) \in \Re^{n\times p}$ is composed of nonlinear functions and $\Theta_i \in \Re^p$ is the vector of p constant parameters. This way, the robot dynamic model (8)-(9) can be expressed as an open-loop error dynamics in terms of a new error coordinate $S_i = \dot{q}_i - \dot{q}_{ri}$. This open-loop error dynamics is useful to design the control law, because it is through the convergence of S_i that the end-effector force and position tasks can be simultaneously achieved. To this end, by adding and subtracting $Y_{ri}\Theta_i$ to (8)-(9) we obtain

$$H_i(q_i)\dot{S}_i + [C_i(q_i,\dot{q}_i) + B_i]S_i = \tau_i + J_{\varphi i}^T(q_i)\lambda_i - Y_{ri}\Theta_i \tag{11}$$

3 Controller Design

3.1 Joint Nominal Reference

According to (6), $\dot{q}_i$ can be decomposed as $\dot{q}_i = Q_{\varphi i}(q_i)\dot{q}_i + J_{\varphi i}^+(q_i)\dot{X}_{Ni}$ [4], [7]. The open-loop error coordinate is given by $S_i = [Q_{\varphi i}(q_i)\dot{q}_i + J_{\varphi i}^+(q_i)\dot{X}_{Ni}] - \dot{q}_{ri}$. Since $\dot{q}_{ri}$ is a velocity-defined variable, it is reasonable to design this reference similarly to $\dot{q}_i$ to build all tracking errors, in order to preserve closed-loop passivity. Then, consider the following definition of the joint nominal reference:

$$\dot{q}_{ri} = Q_{\varphi i}(q_i)(\dot{q}_{di} - \alpha_i\Delta q_i) + J_{\varphi i}^+(q_i)(\dot{S}_f + \beta_i S_f - \dot{X}_{Nj\neq i}) \tag{12}$$

where $\Delta q_i = q_i - q_{di} \in \Re^n$ is the joint position error and $S_f = \Delta\lambda + \int_{t_0}^t \Delta\lambda \, \mathrm{d}t \in \Re$ denotes the force error manifold, for $\Delta\lambda = \lambda - \lambda_d \in \Re$ as the contact force error. Feedback gains $\alpha_i \in \Re^{n\times n}$ and $\beta_i \in \Re$ are a positive diagonal matrix and a positive scalar, respectively. Following the same formulation as for S_i we can say that the force error manifold is given by $S_f = \lambda - \lambda_r$, where $\lambda_r = \lambda_d - \int_{t_0}^t \Delta\lambda \, \mathrm{d}t \in \Re$ is the force nominal reference and $\lambda_d \in \Re$ is the desired contact force profile. Finally, the closed-loop error coordinate is given by:

$$S_i = Q_{\varphi i}(q_i)(\Delta\dot{q}_i + \alpha_i\Delta q_i) - J_{\varphi i}^+(q_i)(\dot{S}_f + \beta_i S_f) \tag{13}$$

[4] Which in fact, maps the equilibrium manifold, as it becomes clear later

The term $\dot{X}_{Nj\neq i}$ in (12) is used to compensate in closed-loop the constrained velocity. This way, (13) is composed of two orthogonal subspaces spanned by $Q_{\varphi i}(q_i)$ and $J_{\varphi i}(q_i)$. Thus, the convergence of S_i implies the independent convergence of position errors $\Delta\dot{q}_i + \alpha_i \Delta q_i$ and force errors $\dot{S}_f + \beta_i S_f$. On the other hand, since $\Delta\dot{q}_i + \alpha_i \Delta q_i$ is mapped onto the tangent subspace, its convergence can only prove position tracking along the tangent direction. Therefore, the following control law must include proper variables to ensure tracking in the normal direction, i.e., to control of the cooperativeness error S_{xi}.

3.2 Control Law

Consider the following model-based control law for the robot arm i:

$$\tau_i = -K_i S_i - J_{\varphi i}^T(q_i)(\lambda_r + \dot{S}_{xi} + \gamma_i S_{xi}) + Y_{ri}\Theta_i \tag{14}$$

where $K_i = K_i^T > 0 \ \in \Re^{n\times n}$ and $\gamma_i > 0 \ \in \Re$ are positive feedback gains. Now, if we substitute (14) into (11), we get the following closed-loop dynamics:

$$H_i(q_i)\dot{S}_i + [C_i(q_i,\dot{q}_i) + \bar{K}_i]S_i + J_{\varphi i}^T(q_i)(\dot{S}_{xi} + \gamma_i S_{xi} - S_f) = \tau_i^* \tag{15}$$

where $\bar{K}_i = B_i + K_i$ and $\tau_i^* = 0$ is a fictitious torque input.

Theorem. [Stable Cooperative Manipulation with Redundant Arms] Consider the robotic system (8)-(9) composed by two constrained redundant arms under the same control law (14) for each arm. The closed-loop robotic system satisfies passivity between the fictitious input torques τ_1^*, τ_2^* and the velocity-defined variables S_1, S_2. Moreover, for each redundant arm, asymptotic convergence for the end-effector force $\Delta\lambda$, for the joint position Δq_i and for cooperativeness S_{xi} tracking errors are achieved. Additionally, a local minimum of a cost function $\Omega_i(q_i)$ is reached by the dynamic reconfiguration of each redundant arm.

Proof. Through the following passivity analysis: $P = \sum_{i=1}^2 S_i^T \tau_i^* = \dot{V} + P_{diss}$, a candidate Lyapunov function V is found for the closed-loop robotic system as follows:

$$\sum_{i=1}^2 S_i^T \tau_i^* = \sum_{i=1}^2 \frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}[S_i^T H_i(q_i)S_i + S_f^2 + S_{xi}^2] + \sum_{i=1}^2 [S_i^T \bar{K}_i S_i + \beta_i S_f^2 + \gamma_i S_{xi}^2]$$

where $V = \sum_{i=1}^2 \frac{1}{2}[S_i^T H_i(q_i)S_i + S_f^2 + S_{xi}^2] \geq 0$ qualifies as a candidate Lyapunov function. Since $\tau_i^* = 0$, thus $\dot{V} = -\sum_{i=1}^2(S_i^T \bar{K}_i S_i + \beta_i S_f^2 + \gamma_i S_{xi}^2) \leq 0$, proving stability. Then the positive definite feedback gains K_i, β_i and γ_i can be employed to modify the transient performance of the system. It is clear that $V(t) \leq V(t_0)$; also, notice that $S_i, S_f, S_{xi} \in \mathcal{L}_\infty$ and $S_i, S_f, S_{xi} \in \mathcal{L}_2$, $\Rightarrow S_i, S_f, S_{xi} \in \mathcal{L}_2 \cap \mathcal{L}_\infty$, and by invoking the Direct Lyapunov Theorem we have that S_i, S_f and S_{xi} converge asymptotically into the equilibrium point, that is $S_i, S_f, S_{xi} \to 0$ as

$t \to \infty$. Notice the explicit convergence of S_{xi}, this stands as the paper major contribution. Finally, to prove the local optimization of $\Omega_i(q_i)$, consider the following: The scalar cost function $\Omega_i(q_i)$ has a local minimum at point q_i^*, if there is a vicinity around q_i^* defined by positive scalar $\epsilon_i > 0$ such that for all points q_i in this vicinity that satisfy $\|q_i - q_i^*\| < \epsilon_i$, the increment of $\Omega_i(q_i)$ has the same sign. If $\Omega_i(q_i) - \Omega_i(q_i^*) \geq 0$, then $\Omega_i(q_i^*)$ is a local minimum. Considering the definition of $\dot{q}_{Ki} = [I - J_i^+(q_i)J_i(q_i)]\frac{\partial \Omega_i(q_i)}{\partial q_i}$, it is evident that $\dot{q}_{Ki}$ moves in the direction on which $\Omega_i(q_i)$ decreases, then $\dot{q}_{Ki}$ vanishes at q_i^*, thus $\Omega_i(q_i)$ is locally optimized. **QED**

4 Simulation Study

Settings. In order to validate the algorithm, a simulation study was carried out using the full dynamic model of a robot-torso, based on the DLR Justin©, with two identical 7 DoF arms manipulating a rigid object.

Manipulating the object. We want to cooperatively move an object along the x-axis, assuming that both arms are already in contact with the object. The desired object pose trajectory is given by: $X_d = [x_d, y_d, z_d, \phi_d, \theta_d, \psi_d] = [0.1\sin(t), -0.1, 0.05, 0, 0, 0]$, where x_d, y_d, z_d (m) are the cartesian coordinates and ϕ_d, θ_d, ψ_d (rad/seg) the euler angles. Simultaneously, the object must be hold with the following exerted force profile $\lambda_d = 100 + 20\tanh(0.1t)$ N.

Redundancy task. Since the desired force profile is increasing with time, the robot kinematic chain is reconfigured in order to protect the weaker wrist joints. This way, redundancy is exploited to overcome the robot joints physical limitations.

Results. On Figure 2 it can be seen that due to the increase in the force profile (arrow size), each redundant arm is dynamically reconfigured in order to satisfy the force requirements. As a consequence of the reconfiguration, the exerted contact force is mainly achieved by the shoulder joints.

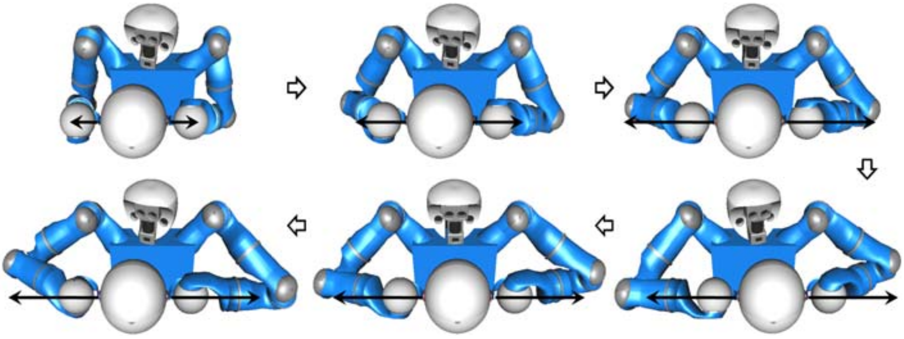


Fig. 2. Kinematic reconfiguration along the object trajectory

5 Conclusions

As the major contribution, our paper proves closed-loop stability in the Lyapunov sense of the cooperativeness error S_{xi}. This result is useful to guarantee simultaneous force-position trajectory tracking of the manipulated object in the constrained (normal) direction, despite the exerted interaction forces. The passivity-based computed-torque like controller has been derived with strict closed-loop stability proofs. Therefore, the extension to other passivity motivated control schemes such as adaptive control, sliding-modes control, cartesian control is straightforward. Notice that since redundancy only reconfigures the kinematic structure without changing the end-effector pose, then it has not direct implication with the convergence of S_{xi}. Therefore, redundancy is here employed to enhance the manipulation capabilities of the system, such as avoiding obstacles, protecting weaker joints, avoiding joint limits, among others.

References

1. Zivanovic, M.D., Vukobratovic, M.K.: Multi-arm cooperating robots. Springer, Heidelberg (2006)
2. Arimoto, S., Liu, Y.H., Naniwa, T.: Model-based adaptive hybrid control for geometrically constrained robots. In: Int. conference on robotics and autom., pp. 618–623 (1993)
3. Nakamura, Y.: Adv. robotics: redundancy and optimization. Addison-Wesley, Reading (1991)
4. Liu, Y., Arimoto, S., Parra-Vega, V., Kitagaki, K.: Decentralized adaptive control of multiple manipulators in cooperations. Int. journal of control, 649–673 (1997)
5. Arimoto, S.: Joint-space orthogonalization and passivity for physical interpretations of dextrous robot motions under geometric constraints. Int. Journal of Robust and Nonlinear Control (1992)
6. Slotine, J.J., Li, W.: On the adaptive control of manipulators. The international journal of robotics research, 49–59 (1987)
7. Navarro-Alarcon, D., Parra-Vega, V., Olguin-Diaz, E.: Minimum set of feedback sensors for high performance decentralized cooperative force control of redundant manipulators. In: Int. workshop on robotic and sensors environments, pp. 114–119 (2008)

A Simple Sample Consensus Algorithm to Find Multiple Models

Carlos Lara-Alvarez, Leonardo Romero,
Juan F. Flores, and Cuauhtemoc Gomez

Division de Estudios de Posgrado
Facultad de Ingenieria Electrica
Universidad Michoacana
Morelia, Mexico
{carlosl,lromero,juanf}@umich.mx, temo@michoacan.gob.mx

Abstract. In many applications it is necessary to describe some experimental data with one or more geometric models. A naive approach to find multiple models consists on the sequential application of a robust regression estimator, such as RANSAC [2], and removing inliers each time that a model instance was detected. The quality of the final result in the sequential approach depends strongly on the order on which the models were. The MuSAC method proposed in this paper discovers several models at the same time, based on the consensus of each model. To reduce bad correspondences between data points and geometric models, this paper also introduces a novel distance for laser range sensors. We use the MuSAC algorithm to find models from 2D range images on cluttered environments with promising results.

1 Introduction

In many applications it is necessary to describe some experimental data with multiple models. A common application in robotics and vision consists on finding models from images. For example, in man–made environments it is useful to discover a set of planes from a set of 3D laser scans. Several approaches to find multiple models have been reported in literature, Franck Dufrenois and Denis Hamad [1] divide those approaches into:

- *Methods that assume the existence of a dominant structure or model in the data set.* These methods successively apply a robust regression estimator. Each time that a model instance is detected, its inliers (data points represented by the model) are removed from the original data.
- *Methods that consider the presence of several structures.* These methods simultaneously extract multiple models from a single data set. Learning strategies range from region growing (or region merging) [3,4,7,6] to probabilistic methods such as Expectation–Maximization [8], or Markov Chain Monte Carlo Methods [5].

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 918–925, 2009.

This paper introduces a simple algorithm to find multiple models from laser scans based on the well known Random Sample Consensus paradigm (RANSAC) [2]. Two similar approaches to the one presented in this paper are MultiRANSAC algorithm proposed by Zuliani and others and Sequential RANSAC (see [11] for more information on both approaches).

MultiRANSAC uses a binary cost function; consequently, it defines the set of data inliers L for a given model θ; that is, $L(\theta) = \{z_i \mid d(z_i, \theta) < t\}$, where $d(\cdot, \cdot)$ is a distance function, and t is a given threshold. To find W models (W is a predefined number), the MultiRANSAC algorithm fuses the W random hypotheses generated at the i–th iteration with the best W models available at the moment to get a new set of W models. Every set of hypotheses $\{\theta_1, \ldots, \theta_W\}$ generated in the MultiRANSAC approach has the property that their corresponding sets of data inliers are pairwise disjoint, $\forall i \neq j \;\; L(\theta_i) \cap L(\theta_j) = \{\}$.

The MuSAC algorithm proposed in this paper has some differences with the MultiRANSAC Algorithm: it does not need to know a priori the number of models, and it can allow those models to have a small amount of common data between them. As a side–effect contribution, a new distance measure is defined. The new distance takes into account how a typical laser range sensor takes measurements from the environment. The rest of the paper is organized as follows. Section 2 reviews the RANSAC method. Section 3 analyzes the Sequential RANSAC approach. Section 4 introduces a new metric used in our approach, called Directional Distance. Section 5 introduces the MuSAC algorithm. Section 6 shows some results of applying the proposed algorithm to solve the problem of obtaining multiple planar models from 2D range images. Finally, the conclusions can be found in Section 7.

2 The Random Sample Consensus Approach

The Random Sample Consensus Approach is very popular for fitting a single model to experimental data. In their seminal paper, Fischler and Bolles[2] describe that the RANSAC procedure is opposite to that of conventional smoothing techniques: rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as the minimum initial data set as feasible and enlarges this set with consistent data, when possible.

The RANSAC strategy has been adopted by many researchers because it is simple and can manage a significant percentage of gross errors (outliers). The smallest number of data points required to uniquely define a given type of model is known as the minimal set (two points define a line, three points define a plane, etc). When using RANSAC, minimal sets are selected randomly; each set produces a hypothesis for the best model. These models are then measured by statistical criteria using all data.

Many efforts have been done to improve the performance of the RANSAC algorithm (replacing the cost function that defines inliers is a usual one). The RANSAC Algorithm optimizes the number of inliers, MSAC (M–Estimator Sample Consensus) [9] incorporates an M–Estimator. M-estimators reduce the effect of outliers by replacing the binary cost function by a symmetric, positive-definite function with a unique minimum at zero. Several functions such as Huber, Cauchy, Tuckey have been proposed. MLeSaC (Maximum Likelihood Sampling Consensus) [10] evaluates the likelihood of the hypothesis, using a Gaussian distribution for inliers and a uniform distribution for outliers.

3 Sequential RANSAC

A naive approach to find multiple models consists of sequentially applying a robust regression estimator, such as RANSAC, and removing inliers each time that a model instance is detected. Figure 1 illustrates the Sequential RANSAC approach. In the first stage (Figure 1(a)), a dominant line is found. After removing the points represented by the first line, it searches for a second line (Figure 1(b)). This process is iterated until the best model does not fulfill a given requirement (usually when the number of points are less than a given threshold). This strategy has some drawbacks: the number of tries is usually large to guarantee that the dominant model can be found on each stage; and, some data points can be misclassified.

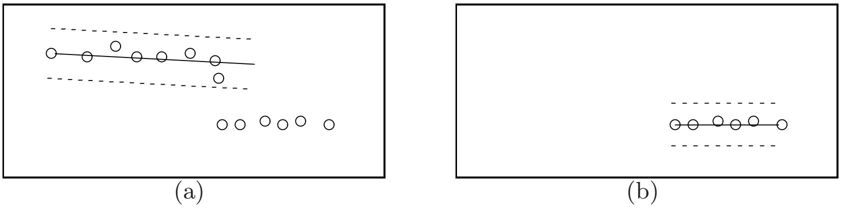


Fig. 1. Using the Sequential RANSAC approach to find multiple lines

4 Directional Distance

The orthogonal distance $d_\perp$ is the preferred metric used to extract geometric models. To increment the rate of points correctly classified, a new metric is proposed. The directional distance $d_\nearrow$ from the data point z_i to the hypothetical model θ_j is defined as

$$d_\nearrow(z_i, \theta_j) = d(z_i, \theta_j \wedge r_i) \tag{1}$$

where $d(\cdot, \cdot)$, is the Euclidean distance between two points; $\theta_j \wedge r_i$, is the intersection point between the model θ_j and the measurement ray r_i associated to z_i (assuming the sensor is at the origin of the reference system).

Suppose we want to evaluate the consensus of the line θ_j as shown in Figure 2. In this case, the observed point z_i (represented by the filled circle) does not correspond to points predicted by θ_j. The intersection point predicted by θ_j (denoted by $\theta_j \wedge r_i$) is represented by the empty circle. $d_\nearrow$ is the distance between the point

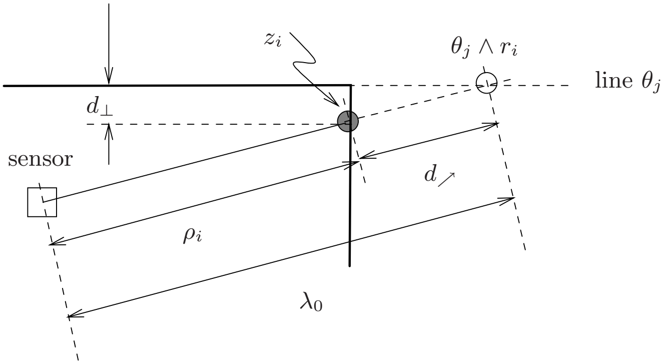


Fig. 2. Comparing the $d_\perp$ distance and the $d_\nearrow$ distance

Z_i and the point $\theta_j \wedge r_i$. Although $d_\perp$ is the best metric for many fitting problems, $d_\nearrow$ is better than $d_\perp$ because it considers the measurement process of laser range sensors. The directional distance represents the real error given the ray r_i on which z_i was measured.

4.1 Point to Line Distance in 2D

A laser range finder takes measurements from the environment by emitting a ray in a given direction α. A ray with direction α represents the set of points (x_1, x_2) given by

$$x_1 = \lambda \cos\alpha; \quad x_2 = \lambda \sin\alpha; \quad \lambda \geq 0 \tag{2}$$

where λ, is the distance from the point (x_1, x_2) to the origin. Let $Ax_1 + Bx_2 + C = 0$ be the equation of a line (a simple model θ) with parameters $[A, B, C]$, and $z_i = (\rho_i, \alpha_i)$ a measured point in polar coordinates. Replacing Equation 2 into the line equation, and solving for the particular value λ_0

$$\lambda_0 = \frac{-C}{B \sin\alpha_i + A \cos\alpha_i} \tag{3}$$

If $\lambda_0 \geq 0$ then the intersection point exists and it defines the distance of the intersection point $\theta \wedge r_i$ to the origin. Finally, $d_\nearrow$ is simply given by

$$d_\nearrow(z_i, \theta_j) = |\lambda_0 - \rho_i|. \tag{4}$$

4.2 Point to Plane Distance in 3D

Analogously to the 2D case, A 3D ray in its parametric form is

$$x_1 = \lambda \sin\alpha \cos\beta, \quad x_2 = \lambda \sin\alpha \sin\beta, \quad x_3 = \lambda \cos\alpha, \quad \lambda \geq 0 \tag{5}$$

where λ is the distance of the point (x_1, x_2, x_3) to the origin. The $3D$ plane equation is $A'x_1 + B'x_2 + C'x_3 + D' = 0$. Let $z_i = (\rho_i, \alpha_i, \beta_i)$ be a point measured at angles α_i, β_i then

$$\lambda_0 = \frac{-D'}{A' \sin\alpha_i \cos\beta_i + B' \sin\alpha_i \cos\beta_i + C' \cos\alpha_i}.$$

and the point to plane distance is

$$d_{\nearrow}(z_i, \theta_j) = |\lambda_0 - \rho_i|. \tag{6}$$

5 The MuSAC Algorithm

The MuSAC Algorithm (Algorithm 1) iterates two phases: hypotheses generation, and selection. The hypotheses generation phase consists of discovering models from the data, and to quantify the relations between the models. The selection stage keeps the best models to the next iteration.

Algorithm 1. MuSAC(Z, the point set; m, a given number of models to be discovered at each iteration; τ, the minimum acceptable consensus)

```
 1: $n \leftarrow 0$, $\Theta = \{\}$
 2: repeat
 3:     for all $i \in \{n+1, \ldots, m\}$ do                ▷ Hypotheses generation
 4:         $\Theta \leftarrow \Theta \cup \{\theta_i\}$, where $\theta_i$ is a random model from $Z$
 5:     end for
 6:     $\forall i, j \in \{1, \ldots, m\}\ c_{ij} \leftarrow \sum_{z \in Z} I(z, \theta_i) I(z, \theta_j)$
 7:     $\Theta^\star \leftarrow \{\}$                                        ▷ Selection
 8:     repeat
 9:         Select $\theta_b \in \Theta$ such that $\forall i \in \{1, \ldots, m\}\ c_{bb} \geq c_{ii}$
10:         if $c_{bb} \geq \tau$ then
11:             $\Theta^\star \leftarrow \Theta^\star \cup \{\theta_b\}$
12:             for all $i \in \{1, \ldots, m\}$, $i \neq b$ do
13:                 if $2c_{ib} > c_{ii}$ then
14:                     $\forall j \in \{1, \ldots, m\}\ c_{ij} \leftarrow 0$
15:                 else
16:                     $\forall j \in \{1, \ldots, m\}\ c_{ij} \leftarrow c_{ij} - c_{bj}$
17:                 end if
18:             end for
19:         end if
20:         $\forall i \in \{1, \ldots m\}\ c_{bi} \leftarrow 0$
21:     until $\forall i \in \{1, \ldots m\}\ c_{ii} < \tau$
22:     $\Theta \leftarrow \Theta^\star, n \leftarrow \mid \Theta^\star \mid$
23: until convergence
24: return $\Theta$
```

Lines 3 through 5 of Algorithm 1 generate a predefined number of hypotheses m. Because the probability that two points belong to the same model is higher when the points are very close to each other, a local strategy is implemented to draw a minimal set of data points: the first point is drawn from the data points without restrictions, and following points are restricted to be in the hypersphere of radius r_m with center in the first point.

To discover hypotheses that correspond to the same object in the environment, the algorithm creates a fully connected graph, where the nodes represent models and the weight of each edge is a statistical measure of the common inliers between the nodes. The graph is represented by Matrix $C = [c_{ij}]_{m \times m}$, where the element c_{ii} represents the consensus (number of inliers) of the model θ_i and the non–diagonal element c_{ij} is a statistical measure of the common inliers between the models θ_i and θ_j, given Z. A large value of c_{ij} indicates a strong relationship between the models θ_i and θ_j; that is, a high plausibility that the models represent the same object in the environment. Analogously, a small value indicates a low plausibility that θ_i and θ_j represent the same object. Matrix C is calculated at line 6 of Algorithm 1, where the function $I(z, \theta_i)$ is defined as

$$I(z,\theta_i) = \begin{cases} \mathrm{1,\ if\ d_{\nearrow}(z,\theta_i) \leq t} \\ \mathrm{0,\ otherwise;} \end{cases}$$

here, t is a predefined threshold.

The selection step is performed in lines 7–21. The model with larger consensus θ_b is selected at line 9. If the consensus of θ_b is greater than a predefined threshold τ, then the consensus of other models is reduced. When $2c_{ib} > c_{ii}$ (line 13) the algorithm considers that models θ_b and θ_i represent the same object in the environment. Then model θ_i is marked as invalid at line 14. Each model is stored in the set $\Theta^{\star}$ (line 11) and the set of these models are the base for the next iteration (line 22).

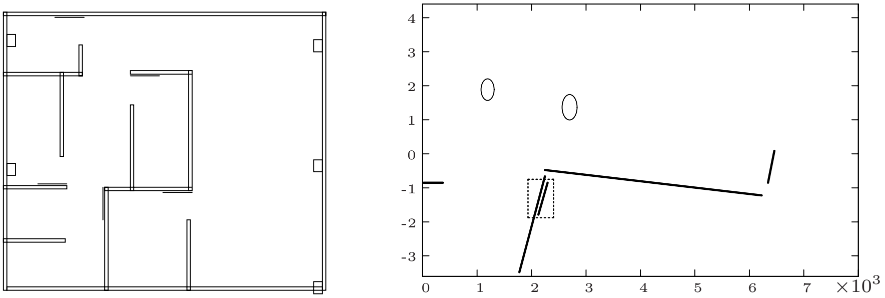
6 Experimental Results

We compare MuSAC with sequential RANSAC and MultiRANSAC algorithms by using the simulated environment shown in Figure 3(a). We generate 1000 random robot poses. A 360^o range scan was taken from each pose, each measurement was corrupted with gaussian noise to each measurement. We use $\tau = 1.96$ std. dev. $\tau = 10$ points, $r_m = 1m$ for all methods; $m = 50$ lines for MuSAC and $W = 20$ lines for MultiRANSAC.

To figure out the performance of algorithms we use S_g y S_b defined as

$$\overline{S_g} = \frac{1}{n}\sum_{v=1}^{n}\frac{g_i}{l_i}, \quad \overline{S_b} = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i}{l_i}$$

where: n, is the number of laser scans; l_i, is the number of lines that generate the i–th scan; g_i, is the number of lines correctly detected, and b_i is the number of lines incorrectly detected. $\overline{S_g}$ is the average ratio of correctly detected models; while $\overline{S_b}$ is the average ratio of spurious models detected.



(a) Simulated Environment

(b) Real Environment

Fig. 3. a. Simulated environment. **b.** Lines extracted from a 2D laser scan in real environment. The dotted rectangle shows two parallel lines correctly discovered, the longer line is a wall while the other one is a door.

Table 1. Results using the simulated environment of Figure 3(a)

Algorithm	$d_\perp$			$d_{\nearrow}$		
	time (msec)	S_g	S_b	time (msec)	S_g	S_b
Sequential RANSAC	7989	0.510	3.440	967	0.956	1.653
MultiRANSAC	6434	0.704	0.186	988	0.773	0.123
MuSAC	3941	0.894	0.601	551	0.907	0.208

Experimental results are shown in Table 1. Sequential RANSAC is the worst because S_b is too high. MuSAC gets a higher value for S_g compared with MultiRANSAC. On the other hand, MultiRANSAC gets a better value for S_b. The average total time used by MuSAC is good enough for real time applications. It is important to note that, the directional distance $d_{\nearrow}$ gets better results than $d_\perp$ in all cases.

For the real test, 2D Laser scans were taken from our laboratory with a laser SICK LMS-200. One challenge for every algorithm is to correctly detect geometric models when they are very close to each other. Figure 3(b) shows a typical result. The dotted rectangle shows that two parallel but different lines were correctly found. Some lines from small objects, such as those marked with ellipses in figure 3(b), were not discovered due to the minimum consensus restriction (τ).

7 Conclusions

The MuSAC algorithm introduced in this paper is faster than MultiRANSAC and it is a good option to extract geometric models from laser scans in real time. The MuSAC algorithm generates a predefined number of random hypotheses from the laser scan and then decides which hypotheses represent the same object in the environment based on their consensus. This paper also introduces the directional distance, this simple metric is helpful for rotational laser sensors because it considers the relationship between the measurement ray and the detected surface. In the near future we want to test our method for other geometric models (such as circles, spheres, etc).

References

1. Dufrenois, F., Hamad, D.: Fuzzy weighted support vector regression for multiple linear model estimation: application to object tracking in image sequences. In: IJCNN, pp. 1289–1294 (2007)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM (1981)
3. Fitzgibbon, A.W., Eggert, D.W., Fisher, R.B.: High-level cad model acquisition from range images. Computer-Aided Design 29, 321–330 (1997)
4. Gorte, B.: Planar feature extraction in terrestrial laser scans using gradient based range image segmentation. In: International Archives of Photogrammetry and Remote Sensing, IAPRS (2007)
5. Han, F., Tu, Z.W., Zhu, S.C.: Range image segmentation by an effective jump-diffusion method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1153 (2004)
6. Jiang, X.Y., Bunke, H.: Fast segmentation of range images into planar regions by scan line grouping. Machine Vision and Applications, 115–122 (1994)
7. Pulli, K., Pietikäinen, M.: Range image segmentation based on decomposition of surface normals. In: Proc. of Scandinavian Conference on Image Analysis (1993)
8. Thrun, S., Martin, C., Liu, Y., fihnel, D., Emery-Muntemerlo, R., Chakrabarti, D., Burgard, W.: A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots (2003)
9. Torr, P.H.S., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. International journal of computer vision (1997)
10. Torr, P.H.S., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78, 2000 (2000)
11. Zuliani, M., Kenney, C.S., Manjunath, B.S.: The multiransac algorithm and its application to detect planar homographies. In: IEEE International Conference on Image Processing (September 2005)

XVI Keynote 5

Pulse Coupled Neural Networks for Automatic Urban Change Detection at Very High Spatial Resolution

Fabio Pacifici[1] and William J. Emery[2]

[1] Department of Computer, Systems and Production Engineering,
Tor Vergata University, Rome, Italy
[2] Department of Aerospace Engineering Science,
University of Colorado at Boulder, CO - USA
f.pacifici@disp.uniroma2.it

Abstract. In this paper, a novel unsupervised approach based on Pulse-Coupled Neural Networks (PCNNs) for image change detection is discussed. PCNNs are based on the implementation of the mechanisms underlying the visual cortex of small mammals and with respect to more traditional neural networks architectures own interesting advantages. In particular, they are unsupervised and context sensitive. The performance of the algorithm has been evaluated on very high spatial resolution QuickBird and WorldView-1 images. Qualitative and more quantitative results are discussed.

Keywords: Change detection, Pulse Coupled Neural Networks, Urban Environment.

1 Introduction

World population growth affects the environment through the swelling of the population in urban areas and by increasing the total consumption of natural resources. Monitoring these changes timely and accurately might be crucial for the implementation of effective decision-making processes. In this context, the contribution of satellite and airborne sensors might be significant for updating land-cover and land-use maps. Indeed, the recent commercial availability of very high spatial resolution visible and near-infrared satellite data has opened a wide range of new opportunities for the use of Earth-observing satellite data. In particular, new systems such as the latest WorldView-1, characterized by the highest spatial resolution, now provide additional data along with very high spatial resolution platforms, such as QuickBird or IKONOS, which have already been operating for a few years.

If on one side this makes available a large amount of information, on the other side, the need of completely automatic techniques able to manage big data archives is becoming extremely urgent. In fact, supervised methods risk to become unsuitable when dealing with such large amounts of data. This is even

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 929–942, 2009.

more compelling if applications dedicated to the monitoring of urban sprawl are considered. In these cases, the big potential provided by very high spatial resolution images has to be exploited for analyzing large areas, which would be unfeasible if completely automatic procedures are not taken into account.

Most of the research carried out so far focused on medium or high spatial resolution images, whereas only few studies have addressed the problem of fully automatic change detection for very high spatial resolution images. In this case, several issues have to be specifically considered. The crucial ones include possible misregistrations, shadow, and other seasonal and meteorological effects which add up and combine to reduce the attainable accuracy in the change detection results.

In this paper, a novel neural network approach for the detection of changes in multi-temporal very high spatial resolution images is proposed. Pulse-coupled neural networks are a relatively new technique based on the implementation of the mechanisms underlying the visual cortex of small mammals. The visual cortex is the part of the brain that receives information from the eye. The waves generated by each iteration of the algorithm create specific signatures of the scene which are successively compared for the generation of the change map. The proposed method is completely automated since analyzes the correlation between the time signals associated to the original images. This means that no pre-processing, except for image registration, is required. Furthermore, PCNNs may be implemented to exploit at the same time both contextual and spectral information which make them suitable for processing any kind of sub-meter resolution images.

This paper is organized as follows. Section 2 recalls the PCNN model. The application of the algorithm is described in Section 3. Final conclusions follow in Section 4.

2 Pulse Coupled Neural Networks

Pulse Coupled Neural Networks entered the field of image processing in the nineties, following the publication of a new neuron model introduced by Eckhorn *et al.* [1]. Interesting results have been already shown by several authors in the application of this model in image segmentation, classification and thinning [2][3], including, in few cases, the use of satellite data [4][5]. Hereafter, the main concepts underlying the behavior of PCNNs are briefly recalled. For a more comprehensive introduction to image processing using PCNN refer to [6].

2.1 The Pulse Coupled Model

A PCNN is a neural network algorithm that, when applied to image processing, yields a series of binary pulsed signals, each associated to one pixel or to a cluster of pixels. It belongs to the class of unsupervised artificial neural networks in the sense that it does not need to be trained. The network consists of nodes with spiking behavior interacting each other within a pre-defined grid. The architecture

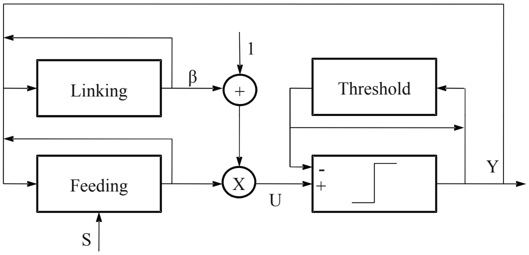


Fig. 1. Schematic representation of a PCNN neuron

of the network is rather simpler than most other neural implementations: there are no multiple layers that pass information to each other. PCNNs only have one layer of neurons, which receives input directly from the original image, and form the resulting *pulse* image.

The PCNN neuron has three compartments. The *feeding* compartment receives both an external and a local stimulus, whereas the *linking* compartment only receives the local stimulus. The third compartment is represented by an active threshold value. When the internal activity becomes larger than the threshold the neuron fires and the threshold sharply increases. Afterwards, it begins to decay until once again the internal activity becomes larger. Such a process gives rise to the pulsing nature of the PCNN.

The schematic representation of a PCNN is shown in Figure 1 while, more formally, the system can be defined by the following expressions:

$$F_{ij}[n] = e^{-\alpha_F} \cdot F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \tag{1}$$

$$L_{ij}[n] = e^{-\alpha_L} \cdot L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \tag{2}$$

where S_{ij} is the input to the neuron (ij) belonging to a 2D grid of neurons, F_{ij} the value of its feeding compartment and L_{ij} is the corresponding value of the linking compartment. Each of these neurons communicates with neighbouring neurons (kl) through the weights given by M and W respectively. M and W traditionally follow very symmetric patterns and most of the weights are zero. Y indicates the output of a neuron from a previous iteration $[n-1]$. All compartments have a memory of the previous state, which decays in time by the exponent term. The constant V_F and V_L are normalizing constants. The state of the feeding and linking compartments are combined to create the internal state of the neuron, U. The combination is controlled by the linking strength, β. The internal activity is given by:

$$U_{ij}[n] = F_{ij}[n] \left\{1 + \beta\, L_{ij}[n]\right\} \tag{3}$$

The internal state of the neuron is compared to a dynamic threshold, Θ, to produce the output, Y, by:

$$Y_{ij}[n] = \begin{cases} 1 & \text{if } U_{ij}[n] > \Theta_{ij}[n] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The threshold compartment is described as:

$$\Theta_{ij}[n] = e^{-\alpha_\Theta} \cdot \Theta_{ij}[n-1] + V_\Theta Y_{ij}[n] \tag{5}$$

where V_Θ is a large constant generally more than one order of magnitude greater than the average value of U.

The algorithm consists of iteratively computing Equation 1 through Equation 5 until the user decides to stop. Each neuron that has any stimulus will fire at the initial iteration, creating a large threshold value. Then, only after several iterations the threshold will be small enough to allow the neuron to fire again. This process is the beginning of the *autowaves* nature of PCNNs. Basically, when a neuron (or group of neurons) fires, an autowave emanates from that perimeter of the group. Autowaves are defined as normal propagating waves that do not reflect or refract. In other words, when two waves collide they do not pass through each other.

PCNNs have several parameters that may be tuned to considerably modify the outputs. The linking strength, β, together with the two weight matrices, scales the feeding and linking inputs, while the three potentials, V, scale the internal signals. The time constants and the offset parameter of the firing threshold can be exploited to modify the conversions between pulses and magnitudes. The dimension of the convolution kernel directly affects the speed of the autowaves. The dimension of the kernel allows the neurons to communicate with neurons farther away and thus allows the autowave to advance farther in each iteration. The pulse behavior of a single neuron is directly affected by the values of α_Θ and V_Θ. The first affects the decay of the threshold value, while the latter affects the height of the threshold increase after the neuron pulses [6].

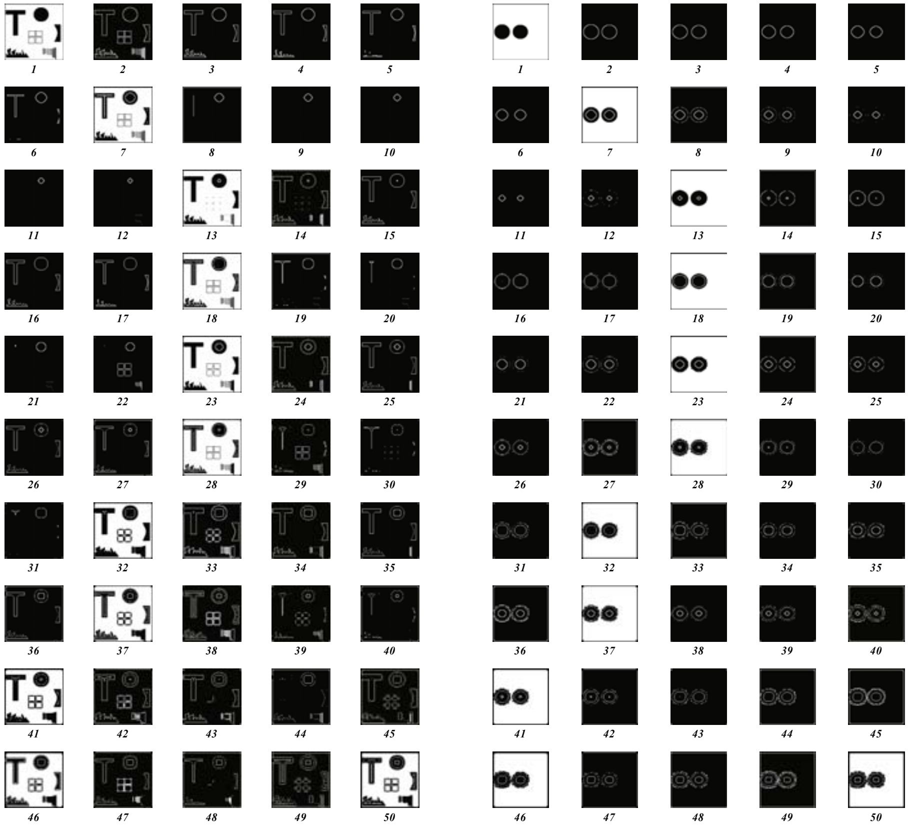
For each unit, i.e. for each pixel of an image, the PCNNs provide an output value. The *time signal* $G[n]$, computed by:

$$G[n] = \frac{1}{N} \sum_{ij} Y_{ij}[n] \tag{6}$$

is generally exploited to convert the pulse images to a single vector of information. In this way, it is possible to have a *global* measure of the number of pixels that fire at epoch $[n]$ in a sub-image containing N pixels. The signal associated to $G[n]$ was shown to have properties of invariance to changes in rotation, scale, shift, or skew of an object within the scene [6].

2.2 Application of PCNN to Toy Examples

PCNNs have been applied to two toy examples to illustrate the internal activity of the model. Figure 2 shows the first 49 iterations of the algorithm with two images of 150 by 150 pixels. The original inputs $(n = 1)$ contain objects with

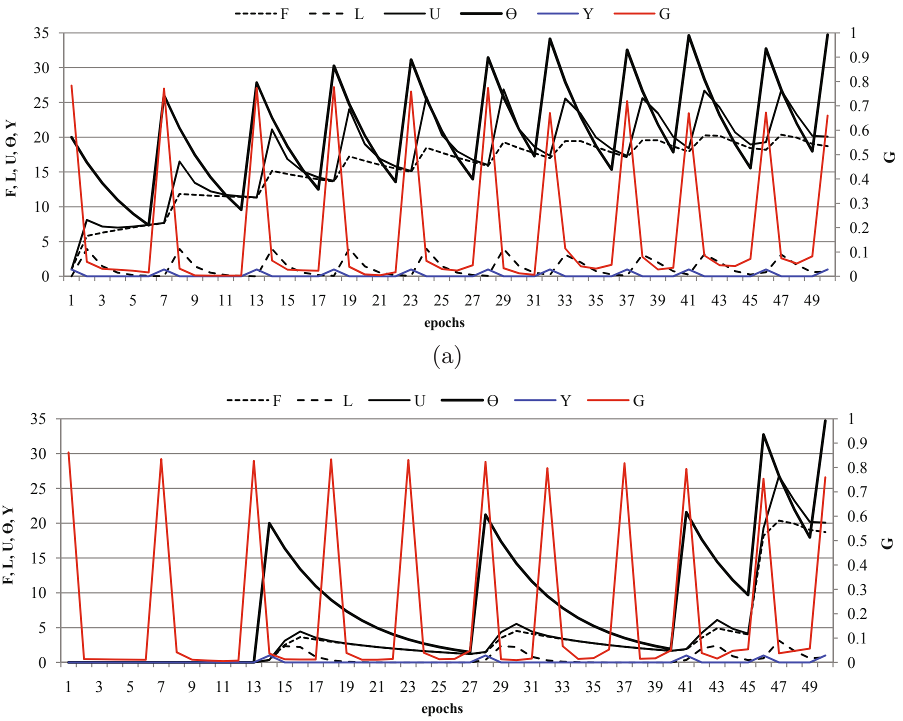


(a) (b)

Fig. 2. Iterations of the PCNN algorithm applied to toy examples of 150 by 150 pixels. As the iterations progress ($n > 1$), the autowaves emanate from the original pulse regions and the shapes of the objects evolve through the epochs due to the pulsing nature of PCNNs.

various shapes, including "T", squares and circles. As the iterations progress ($n > 1$), the autowaves emanate from the original pulse regions and the shapes of the objects evolve through the epochs due to the pulsing nature of PCNNs.

In Figure 3a and Figure 3b are illustrated the progression of the states of a single neuron and trend of G (Equations 1–6) for the toy examples in Figure 2a and Figure 2b, respectively. As shown, the internal activity U rises until it becomes larger than the threshold Θ and the neuron fires ($Y = 1$). Then, the threshold significantly increases and it takes several iterations before the threshold decays enough to allow the neuron to fire again. Moreover, F, L and U maintain values within individual ranges. It is important to note that the threshold Θ reflects the pulsing nature of the *single* neuron, while G gives a *global* measure of the number of pixels that fired at epoch $[n]$.



(a)

(b)

Fig. 3. Progression of the states of a single neuron (in this example, the central pixel) and trend of G for the toy example in Figure 2a and Figure 2b, respectively

3 Change Detection with Pulse-Coupled Neural Networks

The development of fully automatic change detection procedures for very high spatial resolution images is not a trivial task as several issues have to be considered. As discussed in the previous sections, the crucial difficulties include possible different viewing angles, mis-registrations, shadow and other seasonal and meteorological effects which add up and combine to reduce the attainable accuracy in the change detection results. However this challenge has to be faced to fully exploit the big potential offered by the ever-increasing amount of information made available by ongoing and future satellite missions.

PCNNs can be used to individuate, in a fully automatic manner, the areas of an image where a significant change occurred. In particular, the time signal $G[n]$, computed by Equation 6 was shown to have properties of invariance to changes in rotation, scale, shift, or skew of an object within the scene. This last feature makes PCNNs a suitable approach for change detection in very high resolution imagery, where the view angle of the sensor may play an important role.

In particular, the waves generated by the time signal in each iteration of the algorithm create specific signatures of the scene which are successively compared for the generation of the change map. This can be obtained by measuring the similarity between the time signals associated to the former image and the one associated to the latter. A rather simple and effective way to do this is to use a correlation function operating between the outputs of the PCNNs.

The performance of the algorithm was evaluated on different panchromatic satellite sensors, such as QuickBird and WorldView-1. Qualitative and more quantitative results are reported in the rest of this paper.

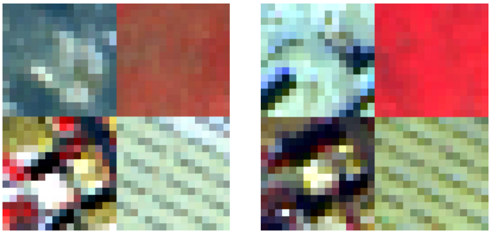
3.1 The Time Signal $G[n]$ in the Multi-spectral Case

To investigate the time signal $G[n]$ on satellite data, two images (2825 × 2917 pixels) acquired by QuickBird on May 29, 2002 and March 13, 2003 over the Tor Vergata University campus (Rome, Italy) have been exploited. Specifically, multi-spectral images (about 2.4m resolution) have been used to have a better comprehension of the PCNN pulsing phase when applied to different bands. In this case, $N = 16 \times 16$ pixels.

Four different conditions shown with false colors in Figure 4 have been considered. In Figure 4a and Figure 4b: (UL) big changes, (UR) change in vegetation cover, (DL) small changes and (DR) no-changes. The first area, big changes, represents the construction of a new commercial building. As shown in Figure 5a, from the very first epochs the pulsing activity of the two images is relatively different, especially if the waveform is concerned. The change in vegetation cover is illustrated in Figure 5b. During the first few epochs, waveform and time dependence of the two signals appear to be similar. For successive epochs, this correlation decreases, especially due to the well known behavior of near infrared band. The time signal for small changes, i.e. when the changed pixels represent a fraction of sub-image considered, is shown in Figure 5c. During the first epochs, waveforms show slight differences, while the time correlation seems to get lost faster than the previous example. Finally, for the no-changes case shown in Figure 5d, it is possible to note that during the initial epochs both the waveform and the time dependence of the two signals appear to be highly correlated.

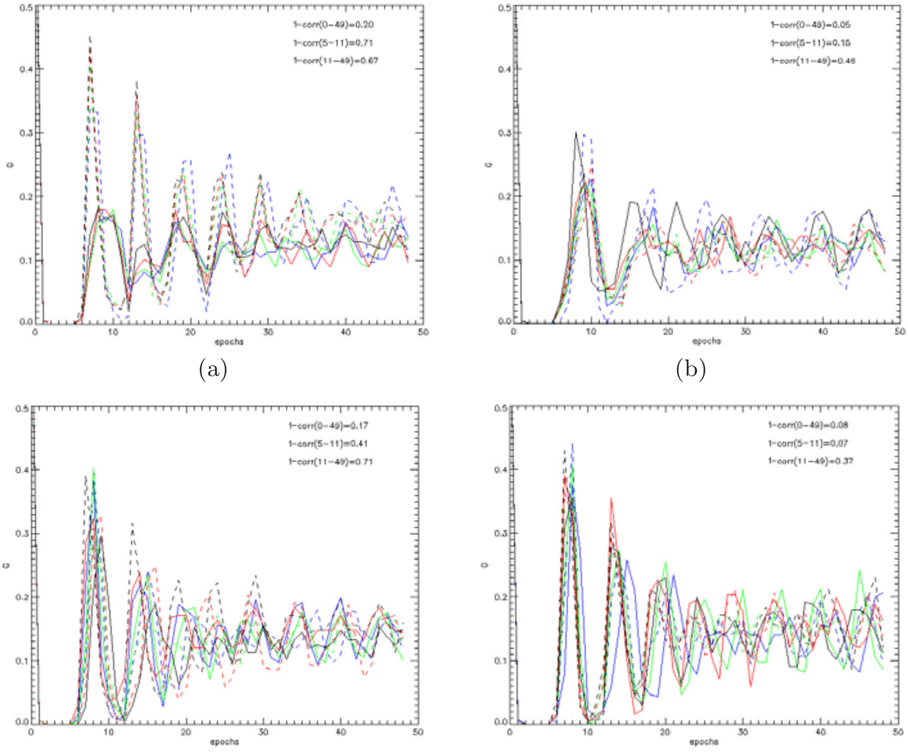
Different values can be obtained considering different epoch intervals. This is concisely expressed in Figure 5, where some correlation values obtained considering specific epochs are reported. In particular, it seems not useful to use a high number of epochs since it is not possible to completely distinguish different land changes. On the other hand, the information derived only from the first oscillation (epochs 5-11) appears to be valuable since it allows the discrimination of various land changes.

From this analysis, it seems that PCNNs, once processing an image pair, might be capable to automatically catch those portions where changes occurred. In such a context, an approach based on *hot spot* detection rather than on changed-pixel detection may be more appropriate given the size of the targets (generally buildings) and the huge volume of data archives that it may be necessary to



(a) (b)

Fig. 4. Multi-spectral QuickBird images (a) 2002 and (b) 2003 shown with false colors: (UL) big changes, (UR) change in vegetation cover, (DL) small changes and (DR) no-changes



(a) (b)

(c) (d)

Fig. 5. The pulsing activity of the two images for the four cases considered in Figure 4. UL case is reported in (a), UR in (b), DL in (c) and DR in (d). Continuous lines represent the 2002 image, while dotted lines correspond to the 2003 image. Red = red channel. Green = green channel, Blue = blue channel, Black = near infrared channel.

analyze in next future. However, the implementation of a pixel based approach with PCNNs is straightforward, using a window sliding one pixel at the time.

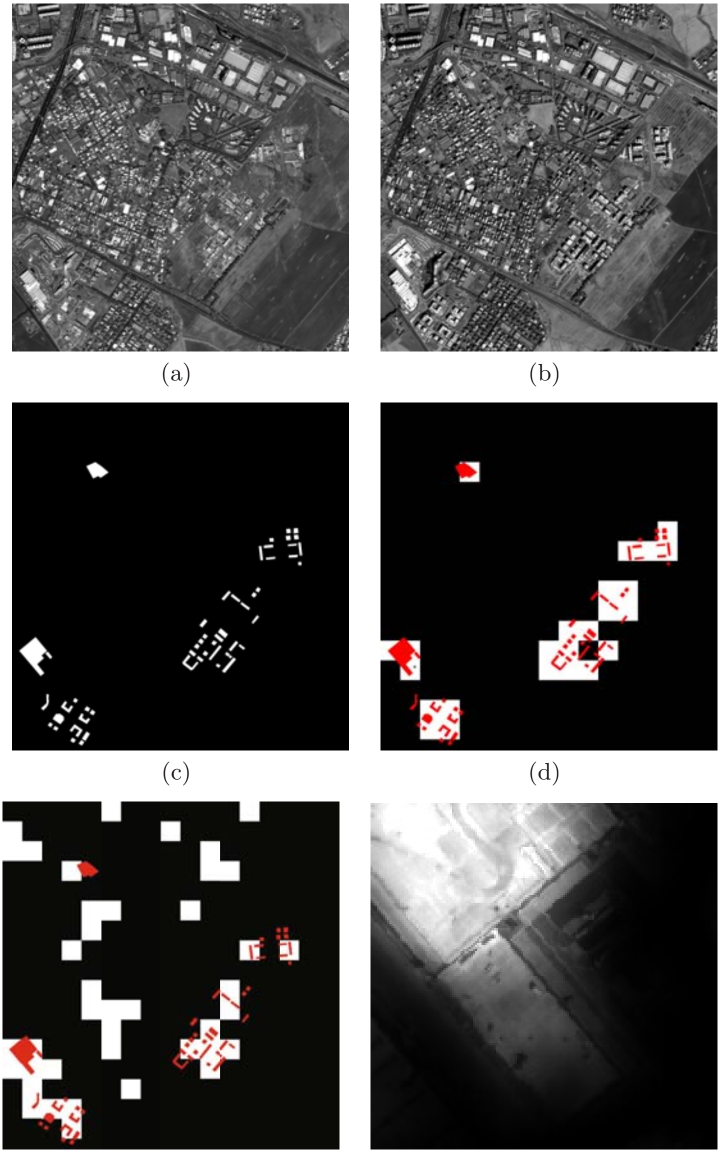
The accuracy of PCNNs in change detection has been evaluated more quantitatively applying PCNNs to the QuickBird imagery of the Tor Vergata University. The panchromatic images have been considered in order to design a single PCNN working with higher resolution (0.6m) rather than 4 different ones processing lower resolution (2.4m) images. The two panchromatic images are shown in Figure 6a and Figure 6b. Few changes occurred in the area during the analyzed time window, the main ones correspond to the construction of new commercial and residential areas. A complete ground reference of changes is reported in Figure 6c. Note that the ground reference included also houses that were already partially built in 2002.

The size of the PCNN was of 100x100 neurons. For the reasons explained previously, it was preferred only to look for the *hot spots* where a change could be rather probable. To operate in this way, the PCNN output values were averaged over of the $10,000$ neurons belonging to 100x100 boxes. An overlap between adjacent patches of 50 pixels (half patch) was considered. Increasing the overlap would have meant to have more spatial resolution out at the price of an increase of the computational burden. Considering this study was aimed at detecting objects of at least some decades of pixels (such as buildings), an overlapping size of 50 pixels was assumed to be a reasonable compromise. The computed mean correlation value was then used to discriminate between changed and not changed area.

The PCNN result is shown in Figure 6d, while in Figure 6e, for sake of comparison, the image difference result is reported. More in detail, in this latter case, an average value was computed for each box of the difference image and a threshold value was selected to discriminate between changed and not changed areas. In particular, the threshold value was chosen to maximize the number of true positives, keeping reasonably low the number of false positives.

What should be firstly noted is that, at least in this application, the PCNN algorithm did not provide any intermediate outputs, with the correlation values alternatively very close to 0 or 1. This avoided to search for optimum thresholds to be applied for the final binary response. The accuracy is satisfactory, as 49 out of the 54 objects appearing on the ground reference were detected with no false-alarms. The missed objects are basically structures that were already present in the 2002 image (e.g. foundations or the first few floors of a building), but not completed yet. On the other side, the result given by the image difference technique, although a suitable threshold value was selected, is rather imprecise, presenting a remarkable number of false alarms.

The image shown in Figure 6f has been obtained by a multi-scale procedure. This consists in a new PCNN elaboration, this time on pixel basis, of one of the hot spots generated with the first elaboration. In particular the change area corresponding to the box indicated by the "$*$" in Figure 6d. It can be noted that the output reported in Figure 6f is more uniformly distributed within the range between 0 and 1. Its value has been multiplied with the panchromatic image

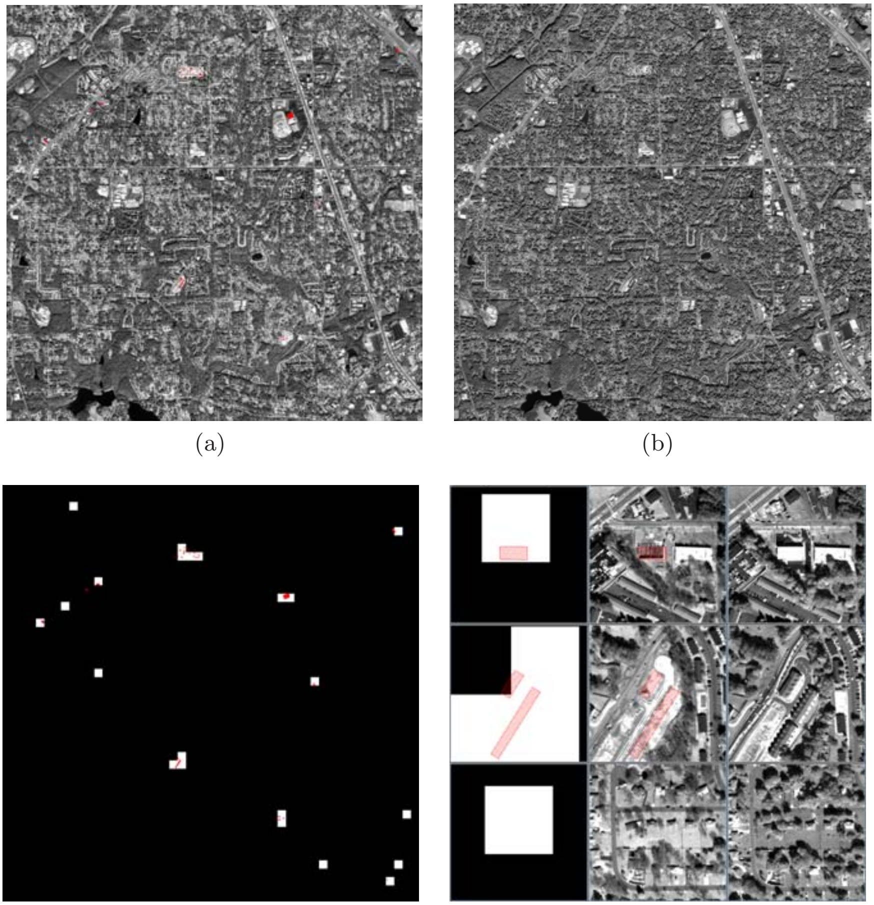


(a) (b)

(c) (d)

(e) (f)

Fig. 6. Panchromatic image of the Tor Vergata University in (a) 2002 and (b) 2003 and (c) the relative ground reference. Change detection result obtained by: (d) the PCNN elaboration and (e) the standard image difference procedure. In (f) is shown the PCNN pixel-based analysis carried out over one of the previously detected changed areas indicated with "∗" in (d).



(a) (b)

(c) (d)

Fig. 7. QuickBird image with ground reference in red (a) and WorldView-1 image (b) of Atlanta. In (c), the change map provided by PCNN and (d) details of the detected hot spots, including a false alarm.

taken in 2003 to have a result which better exploits the very high resolution property of the original image.

3.2 Automatic Change Detection in Data Archives

The study area includes the suburbs of Atlanta, Georgia (U. S. A.). The images were acquired by QuickBird in February 26, 2007 and by WorldView-1 in October 21, 2007 for an approximately extension in area of $25km^2$ (10,000x10,000 pixels). The size of this test case represents an operative scenario where PCNNs give evidence of their potentialities in detecting automatically hot spot areas in data archives. The two images are shown in Figure 7a and Figure 7b, respectively. The ground reference of changes is highlighted in Figure 7a and Figure 7c in red.

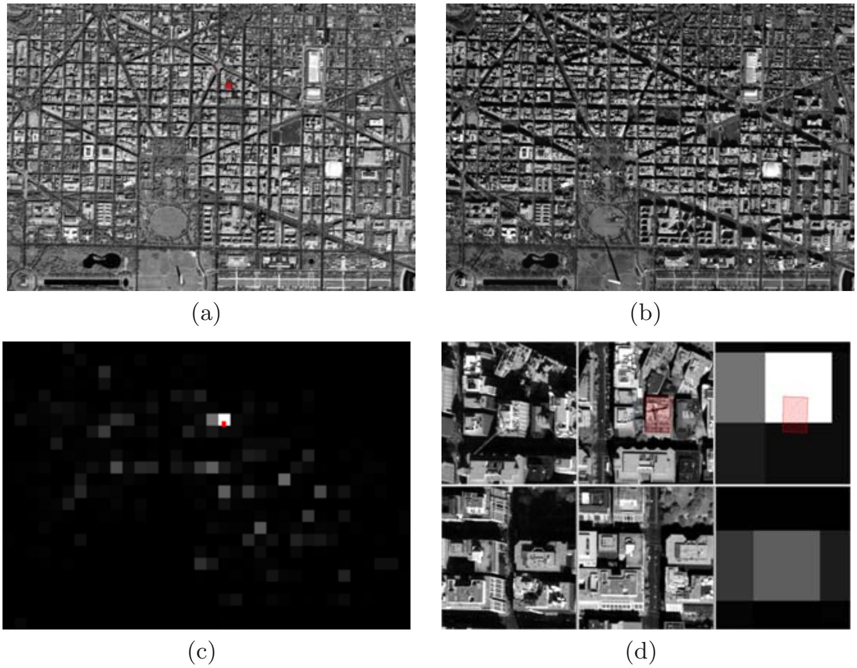


Fig. 8. QuickBird image with ground reference in red (a) and WorldView-1 image (b) of Washington D. C. In (c), the change map provided by PCNN and (d) details of the detected hot spots, including a false alarm.

Note that the ground reference included also houses that were already partially built during the first acquisition. Many changes occurred although the small time window, mainly corresponding to the construction of new commercial and residential buildings.

As shown in Figure 7c, PCNN confirmed to have good capabilities in the automatic detection of the hot spots corresponding to areas which underwent changes, in this case caused from the construction of new structures. For this test case, where the images have comparable viewing angles, PCNNs did not provide any intermediate outputs, with the correlation values alternatively very close to 0 or 1. This avoided to search for optimum thresholds to be applied for the final binary response.

The accuracy is satisfactory, as 30 out of the 34 objects appearing on the ground reference were detected with 6 false alarms, mainly due to presence of leaves on the trees in the WorlView-1 image. The missed objects are basically structures that were already present in the first acquisition (e.g. foundations or the first few floors of a building) but not completed yet, or small isolated houses. Details of the detected hot spots (including a false alarm) are shown in Figure 7d.

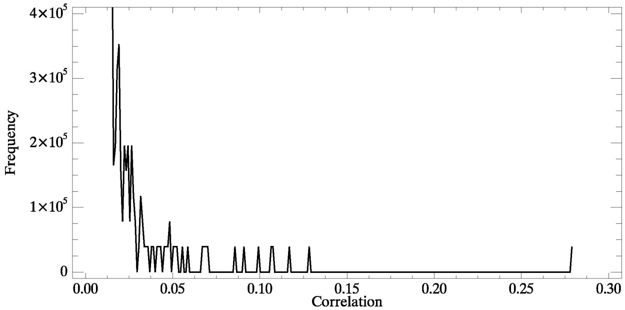


Fig. 9. Frequency of correlation values for the Washington D. C. case. It can be noted that false alarms are characterized by correlation values in the range (0.00; 0.12), while the correlation value of the detected hot spot is more than two times higher, i.e. 0.27.

3.3 Automatic Change Detection in Severe Viewing Conditions

The study area includes the area of Washington D. C. (U. S. A.). The images were acquired by QuickBird in September 23, 2007 and by WorldView-1 in December 18, 2007 for an approximately extension in area of $9km^2$ (7,000x5,000 pixels). In this case, the images have been acquired with very different view angles to investigate the performance of PCNNs in this particularly condition. The images are shown in Figure 8a and Figure 8b, respectively. Only one change occurred in the area due the small time window, corresponding to the demolition of a building (highlighted in red in Figure 8a and Figure 8c).

As shown in Figure 8c, PCNN detected correctly the only hot spot of change. Differently from the previous case, where values were close to 0 or 1, non-changed areas show correlation values slightly bigger than 0. This may be expected due to the very different view angles of the imagery used. For example, the same building is viewed from different directions, occluding different portions of the scene, such as roads or other buildings. However, PCNNs appear to be robust enough to this problem as shown in the plot of Figure 9. Here, on the y-axis, the number of pixels associated to the same measured correlation value is reported. It can be noted that false alarms are characterized by correlation values in the range (0.00; 0.12), while the correlation value of the detected hot spot is more than two times higher, i.e. 0.27. Therefore, in this extreme case, the search for an optimum threshold appear to be straightforward. Details of the detected hot spot and an example false alarm are shown in Figure 8d, respectively.

4 Conclusions

The potential of a novel automatic change detection technique based on PCNNs was investigated. This new neural network model is unsupervised, context sensitive, invariant to an object scale, shift or rotation. Therefore, PCNNs own rather interesting properties for the automatic processing of satellite images.

The approach aiming at discovering changed subareas in the image (the *hot spots*) rather than analyzing the single pixel was here preferred. This might be more convenient when large data sets have to be examined, as it should be the case in the very next years when new satellite missions will be providing additional data along with the ones already available.

The application of PCNNs to sub-meter resolution images of urban areas produced promising results. For the Atalanta area, 30 out of the 34 objects appearing on the ground reference were detected with 6 false alarms, mainly due to presence of leaves on the trees in the WorlView-1 image. The goal of the Washington D. C. scene was to demonstrate the robustness of PCNNs when applied to images acquired with very different view angles. Also in this case, the results were satisfactory since false alarms showed less significant correlation values with respect to real changes.

References

1. Eckhorn, R., Reitboeck, H.J., Arndt, M., Dicke, P.: Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex. Neural Computation 2(3), 293–307 (1990)
2. Kuntimad, G., Ranganath, H.S.: Perfect image segmentation using pulse coupled neural networks. IEEE Transactions on Neural Networks 10(3), 591–598 (1999)
3. Gu, X., Yu, D., Zhang, L.: Image thinning using pulse coupled neural network. Pattern Recognition Letters 25(9), 1075–1084 (2004)
4. Karvonen, J.A.: Baltic sea ice sar segmentation and classification using modified pulse-coupled neural networks. IEEE Transactions on Geoscience and Remote Sensing 42(7), 1566–1574 (2004)
5. Waldemark, K., Lindblad, T., Bečanović, V., Guillen, J.L.L., Klingner, P.: Patterns from the sky satellite image analysis using pulse coupled neural networks for preprocessing, segmentation and edge detection. Pattern Recognition Letters 21(3), 227–237 (2000)
6. Lindblad, T., Kinser, J.M.: Image processing using pulse-coupled neural networks. Springer, Heidelberg (2005)

XVII Intelligent Remote Sensing Imagery Research and Discovery Techniques

Randomized Probabilistic Latent Semantic Analysis for Scene Recognition

Erik Rodner and Joachim Denzler

Chair for Computer Vision
Friedrich Schiller University of Jena
{Erik.Rodner,Joachim.Denzler}@uni-jena.de
http://www.inf-cv.uni-jena.de

Abstract. The concept of probabilistic Latent Semantic Analysis (pLSA) has gained much interest as a tool for feature transformation in image categorization and scene recognition scenarios. However, a major issue of this technique is overfitting. Therefore, we propose to use an ensemble of pLSA models which are trained using random fractions of the training data. We analyze empirically the influence of the degree of randomization and the size of the ensemble on the overall classification performance of a scene recognition task. A thoughtful evaluation shows the benefits of this approach compared to a single pLSA model.

1 Introduction

Building robust feature representations is an important step of many approaches to object recognition. Feature transformation techniques, such as principal component analysis (PCA) or linear discriminant analysis (LDA) offer the possibility to reduce the dimension of an initial feature space using a transformation estimated from all training examples. The main benefit is a compact representation, which exploits that feature vectors in high-dimensional spaces often lie on a lower dimensional manifold.

Within the typical bag-of-features (BoF) approach to image categorization, the reduction of feature vectors using probabilistic Latent Semantic Analysis (pLSA) showed to be beneficial for the overall classification performance [1,2]. The pLSA approach [3] originates from a text categorization scenario, in which a document is represented as an orderless collection of words. With pLSA the representation can be reduced to a collection of latent topics which generate all words of a document. It is natural to transfer this idea to an image categorization scenario and describe an image as a collection or bag of visual words [2]. An estimated distribution of visual word occurrences can be compressed into an image specific distribution of topics. As argued by [4], the pLSA approach has severe overfitting issues. This is due to the number of parameters, which increases with the number of training examples.

In this work, we describe a technique which prevents overfitting by building an ensemble of randomized subspaces and which significantly increase the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 945–953, 2009.

robustness and discriminative power of pLSA reduced features. The basic concept is similar to the random subspace methods of Ho [5] and Rodriguez et al. [6]. Instead of generating an ensemble of classifiers, our approach builds an ensemble of pLSA models which are used for feature transformation. This idea is related to multiple pLSA models used in Brants et al. [7]. Their approach exploits the diversity of generated models due to different random initializations of the EM algorithm which is used to estimate a model. In contrast to that, we generate multiple diverse feature transformations by utilizing the basic idea of Bagging [8] and train each model using a random fraction of the whole data.

Our method can directly be used for the application of scene recognition as described in Bosch et al. [2]. The goal is to categorize an image into a set of predefined scene types, such as mountain, coast, street and forest. Due to the high intraclass variation and low interclass distance, visual words tend to form groups of equal semantic meaning, which can be estimated using pLSA.

The remainder of this paper is structured as follows: The pLSA model and its connections to other approaches are described in Sect. 2. Section 3 presents and discusses our method of generating pLSA-models using a randomization technique. Experimental results within a scene recognition scenario are evaluated in Sect. 4 and show the benefits of our approach. A summary of our findings conclude the paper.

2 Probabilistic Latent Semantic Analysis

A standard approach to image categorization is the bag-of-features (BoF) idea. It is based on the orderless collection of local features extracted from an image and a quantization of these features into V visual words w_j, which build up a visual vocabulary. Images $\{d_i\}_i$ can be represented as a set of histograms $\{c_{ji}\}_i$ which counts how often a visual word w_j is the best description of a local feature in a specific image d_i [2]. Therefore this raw global feature vector associated with an image has as many entries as elements in the visual vocabulary. Especially in the context of scene recognition, it has been shown that the dimensionality reduction of BoF histograms using probabilistic Latent Semantic Analysis (pLSA) leads to performance benefits.

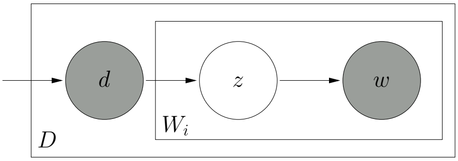


Fig. 1. The asymmetric model of probabilistic Latent Semantic Analysis (pLSA) in plate notation: (observable) visual words w are generated from latent topics z which are specific for each image d (W_i number of visual words, D number of images).

2.1 pLSA Model

The pLSA model, as shown in Fig. 1, models word and image (document) co-occurrences c_{ji} using the joint probability $p(w_j, d_i)$ of a word w_j and an image d_i in the following way:

$$p(w_j, d_i) = p(d_i) \sum_{k=1}^{Z} p(w_j \mid z_k)\, p(z_k \mid d_i). \tag{1}$$

For the sake of brevity, we use the same notation principles as in the original work [3], which abbreviates the event $\mathcal{W} = w_j$ with w_j and skips the formal definition of the random variables $\mathcal{W}, \mathcal{Z}$ and $\mathcal{D}$. Equation (1) illustrates that the pLSA model introduces a latent topic variable $\mathcal{Z}$ and describes all training images as a collection of underlying topics z_k. Note that this model is unsupervised and does not use image labels. By modeling all involved distributions as multinomial, it is possible to directly apply the EM principle to estimate them using visual word counts c_{ji} [3]. Additionally, we can rewrite (1) in matrix notation using $\mathbf{H} = \left[p(w_j, d_i)\right]_{j,i}$, $\mathbf{T} = \left[p(z_k \mid d_i)\right]_{k,i}$, $\mathbf{M} = \left[p(w_j \mid z_k)\right]_{j,k}$ and the diagonal matrix $\mathbf{D} = \left[p(d_i)\right]_{ii}$, which yields:

$$\mathbf{H} = \mathbf{M} \cdot \mathbf{T} \cdot \mathbf{D} \ . \tag{2}$$

This suggests a strong relationship to non-negative matrix factorization (NMF) as introduced by Lee and Seung [9]. In fact, it was highlighted by [10], that NMF of observed values $H_{ji} = c_{ji} \left(\sum_{j'i'} c_{j'i'}\right)^{-1}$ with Kullback-Leibler divergence is equivalent to the pLSA formulation which leads to an instance of the EM principle. In the subsequent sections, we will refer to the matrix $\mathbf{M}$ of topic-specific word probabilities as pLSA model, because it represents the image independent knowledge estimated from the training data.

2.2 pLSA as a Feature Transformation Technique

In [2], the pLSA technique is used as a feature transformation technique, similar to the typical application of PCA. The whole model can be seen as a transformation of BoF histograms $\mathbf{h}^i = \left[H_{ji}\right]_j$ into a new compact Z-dimensional description of each image as a vector of topic probabilities $\mathbf{t}^i = \left[p(z_k|d_i)\right]_k$.

Given an image with an unnormalized BoF histogram $\mathbf{h}$ that is not part of the training set, a suitable feature vector $\mathbf{t}$ has to be found. With a single image, the model equation (2) reduces to $\mathbf{h} = \mathbf{Mt}$ and the estimation of $\mathbf{t}$ can be done by applying the same EM algorithm used for model estimation but without reestimation of the pLSA model (matrix) $\mathbf{M}$. This idea is known as fold-in technique [3] and equivalent to the following NMF-optimization problem:

$$\mathbf{t}(\mathbf{M}, \mathbf{h}) = \underset{\mathbf{t}'}{\operatorname{argmin}}\ \mathrm{KL}(\tilde{\mathbf{h}}, \mathbf{Mt}') \text{ w.r.t. to } \sum_k t'_k = 1 \ , \tag{3}$$

using the normalized BoF histogram $\tilde{\mathbf{h}} = \left(\sum_j h_j\right)^{-1} \mathbf{h}$ and the Kullback-Leibler divergence $\mathrm{KL}(\cdot, \cdot)$.

3 Randomized pLSA

As pointed out by Blei et al. [4], the estimation of the pLSA model leads to overfitting problems. This can be seen by considering the number of parameters involved which grows linearly with the number of training examples. A solution would be to use Latent Dirichlet Allocation [4] which demands sophisticated optimization techniques. In contrast to that, we propose to use an ensemble build by a randomization technique to solve this issue. As opposed to [7], which exploits the diversity of pLSA models resulting from random initializations of the EM-algorithm, we use a randomized selection of training examples, similar to the idea of Random Forests [8] and Random Subspaces [5].

Let $\{\mathbf{M}^l\}_{l=1}^M$ be an ensemble of pLSA models $\mathbf{M}^l = \mathbf{M}(\mathcal{T}^l)$ estimated using a random fraction $\mathcal{T}^l$ of the training data $\mathcal{T}$. We do not select training examples $(\mathbf{h}, y) \in \mathbb{R}^V \times \{1, \ldots, \Omega\}$ of a classification task with Ω classes individually. Instead we propose to select a random fraction of classes $\mathcal{C}^l \subset \{1, \ldots, \Omega\}$ with $|\mathcal{C}^l| = N$ and use all training examples $\mathcal{T}^l = \bigcup_{y_i \in \mathcal{C}^l} \{\mathbf{h}^i\}$ of each selected class . This allows estimating topics which are shared only among a subset of all classes. Each pLSA model $\mathbf{M}^l$ is used to transform BoF histograms $\mathbf{h}^i$ into topic distributions $\mathbf{t}(\mathbf{M}^l, \mathbf{h}^i)$. For training examples in $\mathcal{T}^l$, we use the topic distributions resulting from the pLSA model estimation. All other training examples and each test example are transformed using the “fold-in” technique defined by (3).

One commonly used technique to combine feature transformation models is simply averaging outputs [5] of classifiers trained for each feature set individually. This technique does not allow the classifier to learn dependencies between different models. Therefore we use a concatenation of all calculated feature vectors $\mathbf{t}(\mathbf{M}^l, \mathbf{h}^i)$ as a final feature $\mathbf{t}(\mathbf{h}^i)$:

$$\mathbf{t}(\mathbf{h}^i)^T = \left(\mathbf{t}(\mathbf{M}^1, \mathbf{h}^i)^T, \ldots, \mathbf{t}(\mathbf{M}^M, \mathbf{h}^i)^T\right) \quad . \tag{4}$$

These final feature vectors are of size $M \cdot Z$ and can be used to train an arbitrary classifier. In our experiments, we use an one-vs.-one SVM classifier with a radial basis function kernel.

We have to estimate M pLSA models with the EM algorithm, thus we need roughly M times the computation time of a single model fit. To be exact, we use a fraction of the training data for each model estimation and have to perform the EM algorithm with the “fold-in” technique for each remaining training example:

$$\text{time}_{\text{randomized-plsa}} = \sum_{l=1}^{M} \left(\frac{|\mathcal{T}^l|}{|\mathcal{T}|} \text{time}_{\text{single-model}} + \left(|\mathcal{T}| - |\mathcal{T}^l|\right) \text{time}_{\text{fold-in}} \right) \quad . \tag{5}$$

Therefore we pay for the advantage of reduced overfitting with a higher computational cost.

4 Experiments

We experimentally evaluated our approach to illustrate the benefits of randomized ensembles of pLSA models. In the following, we empirically validate the following hypotheses:

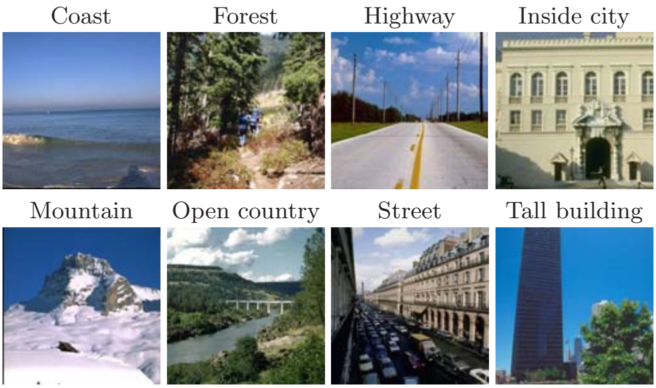


Fig. 2. Example images of each class of the dataset of [11] which we use for evaluation

1. Randomized pLSA ensembles lead to a performance gain in comparision to single pLSA and the usual BoF method, which is most prevalent with a large set of training examples. (Sect. 4.2)
2. With an increasing size M of the ensemble, the recognition rate increases and levels out after a specific size. (Sect. 4.3)
3. The optimal selection of the parameter N (size of the random subset of classes) depends on the size of the training set. (Sect. 4.2)

Additionally, in contrast to other researchers [2], we found that the single pLSA method, in general, does not result in significantly better performance compared to the standard BoF method. A discussion and detailed results of our experiments can be found in Sect. 4.2.

4.1 Experimental Setup

The analysis of the benefits and involved parameters of our method is done using the performance evaluation within a scene recognition scenario. To evaluate our randomized pLSA technique, we use the image dataset of Oliva and Torralba [11], which is a publicly available set of images for evaluating scene recognition approaches [2]. It consists of images from eight different classes which are shown exemplarily in Fig. 2.

All color images are preprocessed as described in [2]. The performance of the overall classification system is measured using unbiased average recognition rates. In contrast to previous work [2], we use Monte Carlo analysis by performing ten independent training and test runs with a randomly chosen training set. This provides us with a statistical meaningful estimate and allows to compare three different approaches: (1) standard BoF without pLSA using normalized histograms (BoF-SVM), (2) a single pLSA model (pLSA) and (3) an ensemble with a varying number of pLSA models (r-pLSA). For the BoF approach directly using BoF histograms **h** as feature vectors, we applied thresholding using mutual information (MI) [12] resulting in a performance gain of 5% for this case.

In all experiments, the number of topics Z is set to 25 and a vocabulary of 1500 visual words is created using the method described in Sect. 4.1. The influence of these parameters was analyzed in previous work [2] and the values showed to be optimal for the dataset of [11].

Feature Extraction. As a local feature representation, we use the OpponentSIFT method proposed in [13]. The task of scene recognition requires the use of information from all parts of the image. Therefore, local descriptors are calculated on a regular grid rather than on interest points only.

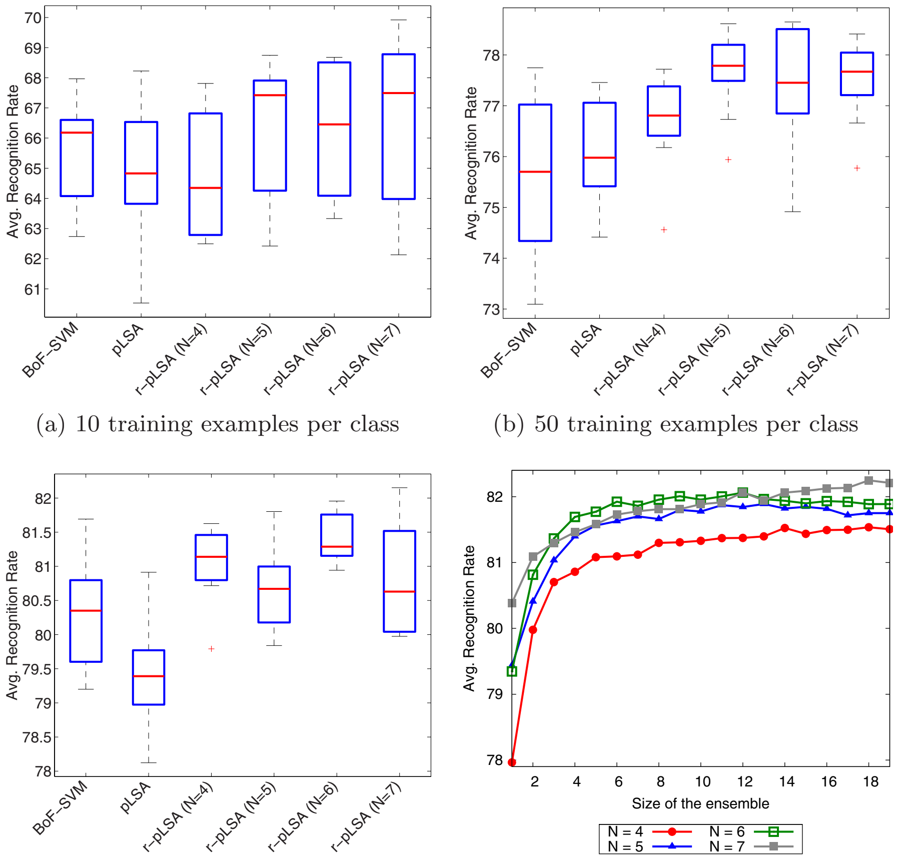
The method of [12], which utilizes a random forest as a clustering mechanism, is used to construct the codebook. It trains a random forest classifier using all local features and corresponding image labels. The leafs of the forest can then be interpreted as individual clusters or visual words. This codebook generation procedure showed superior results compared to standard k-means within all experiments. It also allows us to create large codebooks in a few minutes on a standard personal computer. Note that due to the ensemble of trees, this approach results in multiple visual words for a single local feature. This is not directly modeled by the graphical model underlying pLSA as can be seen in Fig. 1. Nevertheless we can still apply pLSA on the resulting BoF histograms.

4.2 Results and Evaluation

For a different number of training examples (for each class), Figures 3(a) - 3(c) show a comparision of our approach using randomized pLSA ensembles with a standard BoF approach and the utilization of a single pLSA model [2], which is equivalent to randomized pLSA with $N = 8$ and $M = 1$. The classification rates of our approach are displayed for different values of N. To display the results of the multiple training and test runs, we use box plots [14].

At first it can be seen that for nearly all settings (except for 10 training examples and $N = 4$), our randomized pLSA method reaches a higher recognition rate than the usual BoF approach and the method using a single pLSA model [2]. These performance benefits are most prevalent with a large number of training examples. Another surprising fact is that the method proposed by [2] is not significantly better than the simple BoF method. This might be due to our use of MI-thresholding for raw BoF histograms. Another reason could be the analysis using fixed training and test sets in the comparision performed by [2], which does not lead to significant results. With a glance at the box plots for different values of N, we can see that it is hard to determine an optimal parameter value. However a value of $N = 5$ seems to be a reasonable choice.

Note that the absolute recognition performance of $81 - 82\%$ for 150 examples is lower than the best values obtained by [2], which are 87.8% on a test set and 91.1% on a validation set. This is mainly due to different local features and the incorporation of spatial information, which we do not investigate in this paper. However, our idea of randomized pLSA ensembles could be well adopted to use spatial pyramids as proposed in [2].



(a) 10 training examples per class

(b) 50 training examples per class

(c) 150 training examples per class

(d) Influence of the ensemble size

Fig. 3. Evaluation using average recognition rate of the whole classification task: (a-c) Comparision of a usual BoF approach (BoF-SVM), pLSA reduced features and our approach utilizing a randomized ensemble of multiple pLSA models (r-pLSA) using training examples from $N = 4, 5, 6, 7$ random classes. The median of the values is shown by the central mark, top and bottom of the box are the 0.25 and 0.75 percentiles, the whiskers extend to the maximum and minimum values disregarding outliers, and outliers are plotted individually by small crosses [14]. 3(d) classification performance of r-pLSA with a varying size of the ensemble for a fixed training and test set.

4.3 Influence of the Ensemble Size

As can be seen from Fig. 3(d), increasing the number M of pLSA models yields a better overall performance. As expected this leads to convergence after a specific size of the ensemble. A similar effect of the ensemble size can be observed when using Random Forests [8]. Because of the ability of the SVM classifier to build maximum margin hypotheses, the effect of overfitting due to an increasing number of features, and thus to an increasing VC dimension, does not occur.

5 Conclusion and Further Work

We showed that utilizing a randomization principle, an ensemble of pLSA models can be build, which offers a feature transformation technique that is not prone to overfitting compared to a single pLSA model. In a scene recognition scenario, this technique leads to a better recognition performance in comparision with a single model or a standard bag-of-features approach. Our experiments also showed that the recognition performance increases with more pLSA models and levels out. An interesting possibility for future research would be to study ensembles of models estimated with Latent Dirichlet Allocation, which is a more sophisticated method for topic discovery and a well-known Bayesian method [4]. Finally, experiments should be performed using other datasets with more classes and analyzing the trade-off between a better recognition rate and a higher computational cost.

References

1. Quelhas, P., Monay, F., Odobez, J.M., Gatica-Perez, D., Tuytelaars, T., Van Gool, L.: Modeling scenes with local descriptors and latent aspects. In: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 883–890 (2005)
2. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. IEEE Trans. Pattern Anal. Mach. Intell. 30(4), 712–727 (2008)
3. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 42(1-2), 177–196 (2001)
4. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
5. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20(8), 832–844 (1998)
6. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. 28(10), 1619–1630 (2006)
7. Brants, T., Chen, F., Tsochantaridis, I.: Topic-based document segmentation with probabilistic latent semantic analysis. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 211–218 (2002)
8. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
9. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems, vol. 1998, pp. 556–562. MIT Press, Cambridge (2001)
10. Gaussier, E., Goutte, C.: Relation between plsa and nmf and implications. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 601–602 (2005)

11. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision 42(3), 145–175 (2001)
12. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Advances in Neural Information Processing Systems, pp. 985–992 (2006)
13. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluation of color descriptors for object and scene recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
14. Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley, Reading (1977)

Object Contour Tracking Using Foreground and Background Distribution Matching

Mohand Saïd Allili*

Université du Québec en Outaouais,
Département d'Informatique et d'Ingénierie,
101 Rue St-Jean-Bosco, Local: B-2022,
Gatineau, Québec, J8X 3X7
Tel.: +1 (819) 595 3900 ext. 1601
mohandsaid.allili@uqo.ca

Abstract. In this paper, we propose an effective approach for tracking distribution of objects. The approach uses a competition between a tracked objet and background distributions using active contours. Only the segmentation of the object in the first frame is required for initialization. The object contour is tracked by assigning pixels in a way that maximizes the likelihood of the object versus the background. We implement the approach using an EM-like algorithm which evolves the object contour exactly to its boundaries and adapts the distribution parameters of the object and the background to data.

1 Introduction

Object tracking using deformable models is a very important research field in computer vision and image processing, and it has a variety of applications, such as video surveillance, video indexing and retrieval, robotics, etc. Recently, several approaches tackled this problem using foreground (object) distribution matching [1,3,4,6,7]. Those approaches track an object in each frame of the video by trying to find the region in the frame whose interior generates a sample distribution over a relevant variable (target object model) which has the best match with the reference model distribution. Such an approach has the advantage that no motion model needs to be fitted for the tracked objects. However, it has two major limitations. First, the tracking becomes very sensitive to both initial curve positions and model distribution, which may converge the object contour to incorrect local optima [4,7]. Second, the appearance of an object may slightly vary over time (e.g. due to illumination changes or viewing geometry); therefore, the basic assumption of the approach -similarity between the reference and target object appearance- will no longer be valid [1].

To illustrate the aforementioned limitations, Fig. (1) shows two examples where tracking using foreground matching fails. In the first example (first row), the initial curve used to track the object (the shaded disc surrounded by the

* We thank the Université du Québec en Outaouais for the Start up grant.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 954–961, 2009.

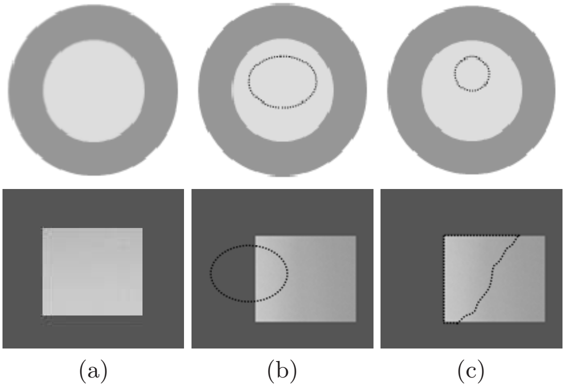


Fig. 1. Example where foreground matching-based tracking fails. The target object is the shaded disc surrounded by a ring in the first row, and the small shaded rectangle in the second row. In each row, (a) represent the reference object model. The dashed curve in (b) and (c) represent the initial and final position of the curve, respectively.

ring) is inside the object. Since the sample distribution at each point of the curve exceeds the reference model distribution, the curve would shrink and ultimately disappear. In the second example (second row), the shading of the object (the rectangle inside) is altered because of an illumination change. Consequently, the curve did not capture the whole object.

Int this paper, we propose a flexible model for object tracking based on variational curve evolution. The *foreground matching* for tracking is augmented with *background matching* (or object-background mismatching) force, which avoids undesirable local optima and augments the tracking accuracy. In addition, the proposed model allows to adapt the distribution of each tracked object and the background to appearance changes using an EM-like approach. We show the effectiveness of the proposed model on tracking examples using real-world video sequences.

This paper is organized as follows: Section (2) presents the proposed model for object tracking. Section (3) presents some experiments that validate the model. Finally, we end with a conclusion and some future work perspectives.

2 The Proposed Model

Let $\Omega \subset \mathbb{Z}^+ \times \mathbb{Z}^+$ be the domain of the image and R_o be the area of the object to be tracked through an image sequence. We suppose the sequence is composed of the frames I_ℓ where $0 \leq \ell < \infty$. The image data $\mathcal{D}$ can be real-valued, such as image intensity, or vector-valued, such as color or texture features. In our case, we use a feature vector $I(\mathbf{x}) = (u_1(\mathbf{x}), ..., u_d(\mathbf{x}))$ that combines color and texture characteristics, where $\mathbf{x}$ represents the pixel coordinates (x, y).

To represent the distribution of high-dimensional image data, the histogram is not the optimal choice since the data are generally very sparse. Therefore, we choose a parametric representation. Let M_ℓ (resp. $\bar{M}_\ell$) and $M_{\ell+1}$ (resp. $\bar{M}_{\ell+1}$) be two parametric mixture models that characterize the object (resp. the background) in two consecutive frames I_ℓ and $I_{\ell+1}$. We denote by $\boldsymbol{\Theta}_\ell$ (resp. $\bar{\boldsymbol{\Theta}}_\ell$) and $\boldsymbol{\Theta}_{\ell+1}$ (resp. $\bar{\boldsymbol{\Theta}}_{\ell+1}$) the mixture parameters of the object (resp. the background) in those frames, respectively. The mixture parameters are estimated initially in the frame I_0 using the maximum likelihood estimation, which is obtained by minimizing the following functions:

$$\boldsymbol{\Theta}_0 = \text{argmin}_{\boldsymbol{\Theta}} \left(E(\boldsymbol{\Theta}) = -log\Big(\mathcal{L}(R_o, \boldsymbol{\Theta})\Big)\right) \tag{1}$$

and:

$$\bar{\boldsymbol{\Theta}}_0 = \text{argmin}_{\bar{\boldsymbol{\Theta}}} \left(E(\bar{\boldsymbol{\Theta}}) = -log\Big(\mathcal{L}(\bar{R}_o, \bar{\boldsymbol{\Theta}})\Big)\right) \tag{2}$$

where $\bar{R}_o$ designates the complement of the object to the background, and $\mathcal{L}(R_o, \boldsymbol{\Theta})$ and $\mathcal{L}(\bar{R}_o, \bar{\boldsymbol{\Theta}})$ are given by:

$$\mathcal{L}(R_o, \boldsymbol{\Theta}) = \prod_{\mathbf{X} \in R_o} \left(\sum_{k=1}^{K} \pi_k p(I(\mathbf{x})|\theta_k) \right) \tag{3}$$

$$\mathcal{L}(\bar{R}_o, \bar{\boldsymbol{\Theta}}) = \prod_{\mathbf{X} \in \bar{R}_o} \left(\sum_{h=1}^{\bar{K}} \bar{\pi}_h p(I(\mathbf{x})|\bar{\theta}_h) \right). \tag{4}$$

where $(\theta_k, \pi_k)_{k=1,\ldots,K}$ and $(\bar{\theta}_h, \bar{\pi}_h)_{h=1,\ldots,\bar{K}}$ designate the parameters of the object and the background mixture models, respectively.

We suppose that the object contour is initialized manually in the first frame of the sequence. Given the position, the distribution and the contour of the object in the frame I_ℓ, we aim to track the object boundaries in the frame $I_{\ell+1}$ based on curve evolution. In what follows, we denote the evolved object contour by $\boldsymbol{\gamma}$. To maximize between frames *foreground* and *background matching*, we propose to minimize the following energy functional:

$$J(\boldsymbol{\gamma}, \boldsymbol{\Theta}_{\ell+1}, \bar{\boldsymbol{\Theta}}_{\ell+1}) = \left\{ \big[E(\boldsymbol{\gamma}, \boldsymbol{\Theta}_{\ell+1}) - E(\boldsymbol{\Theta}_\ell)\big] + \big[E(\boldsymbol{\gamma}, \bar{\boldsymbol{\Theta}}_{\ell+1}) - E(\bar{\boldsymbol{\Theta}}_\ell)\big] \right\} \tag{5}$$

where the energies E are those defined in Eqs. (1) and (2). Using the same manipulation that we used in [1], we can demonstrate that, by using Jensen inequality [5], functional (5) leads to the following inequalities:

$$E(\boldsymbol{\gamma}, \boldsymbol{\Theta}_{\ell+1}) \leq E(\boldsymbol{\Theta}_\ell) + \iint_{R'_o} \mathcal{Q}_1(\mathbf{x}, \boldsymbol{\Theta}_{\ell+1}) d\mathbf{x} \tag{6}$$

$$E(\boldsymbol{\gamma}, \bar{\boldsymbol{\Theta}}_{\ell+1}) \leq E(\bar{\boldsymbol{\Theta}}_\ell) + \iint_{\bar{R}'_o} \mathcal{Q}_2(\mathbf{x}, \bar{\boldsymbol{\Theta}}_{\ell+1}) d\mathbf{x} \tag{7}$$

where R'_o designates the region delimited by the evolved curve $\boldsymbol{\gamma}$ in the frame $I_{\ell+1}$, and $\bar{R}'_o$ designates its complement in the same frame. The terms $\mathcal{Q}_1(\mathbf{x}, \boldsymbol{\Theta}_{\ell+1})$ and $\mathcal{Q}_2(\mathbf{x}, \bar{\boldsymbol{\Theta}}_{\ell+1})$ are given by:

$$\mathcal{Q}_1(\mathbf{x}, \boldsymbol{\Theta}_{\ell+1}) = -\sum_{k=1}^{K} p(\theta_k|I(\mathbf{x}))\log\left(\frac{\pi'_k p(I(\mathbf{x})|\theta'_k)}{\pi_k p(I(\mathbf{x})|\theta_k)}\right) \tag{8}$$

$$\mathcal{Q}_2(\mathbf{x}, \bar{\boldsymbol{\Theta}}_{\ell+1}) = -\sum_{h=1}^{\bar{K}} p(\bar{\theta}_h|I(\mathbf{x}))\log\left(\frac{\bar{\pi}'_h p(I(\mathbf{x})|\bar{\theta}'_h)}{\bar{\pi}_h p(I(\mathbf{x})|\bar{\theta}_h)}\right) \tag{9}$$

where (θ_k, π_k) and $(\theta'_k, \pi'_k), k = 1, ..., K$, (resp. $(\bar{\theta}_h, \bar{\pi}_h)$ and $(\bar{\theta}'_h, \bar{\pi}'_h), h = 1, ..., \bar{K}$) are the object (resp. background) mixture parameters in the frames I_ℓ and $I_{\ell+1}$, respectively. Given that the energies $E(\boldsymbol{\gamma}, \boldsymbol{\Theta}_{\ell+1})$ and $E(\boldsymbol{\gamma}, \bar{\boldsymbol{\Theta}}_{\ell+1})$ are lower-bounded, respectively, by $(\boldsymbol{\Theta}_\ell)$ and $E(\bar{\boldsymbol{\Theta}}_\ell)$, and upper-bounded according to Eqs. (6) and (7), then minimizing them amounts to minimize the integrals in the right hand sides of Eqs. (6) and (7).

In the final step of the proposed model, we couple the region with boundary information of the image to allow for good alignment of the curve $\boldsymbol{\gamma}$ with strong discontinuities of the image. To this end, we minimize the following term:

$$J_b(\boldsymbol{\gamma}) = \oint_0^{L(\boldsymbol{\gamma})} \varphi\left(\mathbf{P}(s)\right) ds \tag{10}$$

where s denotes the arc-length parameter and $L(\boldsymbol{\gamma})$ is the length of the curve $\boldsymbol{\gamma}$. Finally, φ designates a strictly decreasing function of the boundary plausibility $\mathbf{P}(s)$, which is given by $\varphi(\mathbf{P}(s)) = \frac{1}{1+\mathbf{P}(s)}$. The boundary plausibility is calculated using the method proposed in [2]. Minimizing (10) aligns the contour $\boldsymbol{\gamma}$ with high discontinuities of color and texture features in the image while keeping the curve smooth during its evolution.

The minimization of the coupled energy functional according to $\boldsymbol{\gamma}$, $\boldsymbol{\Theta}_{\ell+1}$ and $\bar{\boldsymbol{\Theta}}_{\ell+1}$ is achieved using Euler-Lagrange Equations, which are resolved using the steepest descent method. To allow for automatic topology changes for the object contour, due to occlusions for example, we propose to use the level set formalism [9]. In this formalism, the evolved curve $\boldsymbol{\gamma}$ is embedded as a zero level set of a distance function $\Phi : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. Then, $\boldsymbol{\gamma} = \{\mathbf{x}/\Phi(\mathbf{x}) = 0\}$, where we use the fact that $\Phi(\mathbf{x}) < 0$ if $\mathbf{x}$ is inside the curve $\boldsymbol{\gamma}$ and $\Phi(\mathbf{x}) > 0$ if $\mathbf{x}$ is outside the curve. The final motion equation of the zero level set is given as follows:

$$\begin{aligned}\frac{\partial \Phi(\mathbf{x}, t)}{\partial t} = &\Bigg\{\alpha\Big[\varphi(\Phi)\kappa + \nabla\varphi(\Phi)\cdot\nabla\Phi\Big] \\ + \quad &\beta\Bigg[\sum_{k=1}^{K} p(\theta_k|I(\mathbf{x}))\log\left(\frac{\pi'_k p(I(\mathbf{x})|\theta'_k)}{\pi_k p(I(\mathbf{x})|\theta_k)}\right) \\ + \quad &\sum_{h=1}^{\bar{K}} p(\bar{\theta}_h|I(\mathbf{x}))\log\left(\frac{\bar{\pi}'_h p(I(\mathbf{x})|\bar{\theta}'_h)}{\bar{\pi}_h p(I(\mathbf{x})|\bar{\theta}_h)}\right)\Bigg]\Bigg\}\|\nabla\Phi\|\end{aligned} \tag{11}$$

In the above equation, κ stands for the curvature of the zero level set function. The constants α and β are used to control the contribution of the boundary and region information.

Finally, the minimization of the coupled energy functionals (5) and (10) allows for the mixture models of the object and the background to be adapted to data. For this goal, we assume mixtures of multivariate Gaussian distributions for both the object and the background models. Therefore, the minimization of the coupled functional according to mixture parameters is performed in an EM-like algorithm, which leads to the following updating equations:

$$\mu'_k = \frac{\iint_{R'_o} t_k I(\mathbf{x}) d\mathbf{x}}{\iint_{R'_o} t_k d\mathbf{x}} \tag{12}$$

$$\varSigma'_k = \frac{\iint_{R'_o} t_k \left[(I(\mathbf{x}) - \mu'_k)\right] \left[(I(\mathbf{x}) - \mu'_k\right]^T d\mathbf{x}}{\iint_{R'_o} t_k d\mathbf{x}} \tag{13}$$

$$\pi'_k = \frac{\iint_{R'_o} t_k d\mathbf{x}}{\iint_{R'_o} d\mathbf{x}} \tag{14}$$

$$\bar{\mu}'_h = \frac{\iint_{\bar{R}'_o} t_h I(\mathbf{x}) d\mathbf{x}}{\iint_{\bar{R}'_o} t_h d\mathbf{x}} \tag{15}$$

$$\bar{\varSigma}'_h = \frac{\iint_{\bar{R}'_o} t_h \left[(I(\mathbf{x}) - \bar{\mu}'_h)\right] \left[(I(\mathbf{x}) - \bar{\mu}'_h\right]^T d\mathbf{x}}{\iint_{\bar{R}'_o} t_h d\mathbf{x}} \tag{16}$$

$$\bar{\pi}'_h = \frac{\iint_{\bar{R}'_o} t_h d\mathbf{x}}{\iint_{\bar{R}'_o} d\mathbf{x}} \tag{17}$$

where $t_k = p(\theta_k|I(\mathbf{x})) = \frac{\pi_k p(I(\mathbf{X})|\theta_k)}{\sum_{j=1}^{K} \pi_j p(I(\mathbf{X})|\theta_j)}$ and $t_h = p(\bar{\theta}_h|I(\mathbf{x})) = \frac{\bar{\pi}_h p(I(\mathbf{X})|\bar{\theta}_h)}{\sum_{l=1}^{\bar{K}} \bar{\pi}_l p(I(\mathbf{X})|\bar{\theta}_l)}$. The final algorithm for tracking is summarized as follows:

```
Algorithm:
1- Initialize the object in the first frame $I_0$.
2- For each new frame $I_{\ell+1}$ $(0 \leq \ell < \infty)$:
    While (the object contour has not converged) do
    {
        Evolve the object contour using Eq. (11).
        Update the object and background mixture
        parameters using Eqs. (12) to (17).
    } End while.
```

The convergence of the level sets is detected when:

$$\mathrm{Max}_{(\varPhi(\mathbf{x},t)=0)} \left(|\varPhi(\mathbf{x}, t+1) - \varPhi(\mathbf{x}, t)| \right) < \xi \tag{18}$$

where ξ is a predefined threshold. The above criterion means that contour convergence is reached when the maximum change in the zero level set between two successive iterations, t and $t+1$ using Eq. (11), is below the threshold ξ.

3 Experiments

In our experiments, we compared the proposed model with the approach in [7] which uses foreground matching and active contours for tracking. In the conducted tests, we used videos from the Wallflower database. We used the texture features that we developed in [2] which are combined with color features to build the vector $I(\mathbf{x})$. Finally, we set experimentally the parameters α and β to 0.5 and ξ to 0.5 in Eq. (18).

In the example shown in Fig. (2), the target object is the walking person. The video contains 1744 frames and the tracking was performed from frame 1509 to 1935. However, since the object is not completely visible in the first frame, only the visible part is used to calculate the reference model (see the first frame). Since *foreground matching* tracks only the part corresponding to the reference foreground model, a part of the object was missed. That is, the contour did not adapt to the new distribution of the object. Our model cured this problem thanks to the *background matching* force that acted simultaneously with *foreground matching* to align the contour with the real object boundaries. Fig. (3) shows a tracking example where the target object undergoes an illumination change. The video contains 1744 frames and the tracking was performed from frame 1398 to 1498. The graphs in the same figure show the color scatter distribution of the frames. We can observe the change in the appearance of the frames due to illumination change. The model in [7] failed to find the correct object boundaries. Our model improved the accuracy of tracking where the major part of the object was correctly located in most of the frames.

To measure quantitatively the accuracy of tracking, we hand-segmented the objects in the shown examples, which we consider as ground truth, and we compared the tracking results using the following criterion [8]:

$$\mathcal{E}_\ell = \frac{\left|\left(G_o^\ell - R_o^\ell\right) \cup \left(R_o^\ell - G_o^\ell\right)\right|}{|G_o^\ell| + |R_o^\ell|} \tag{19}$$

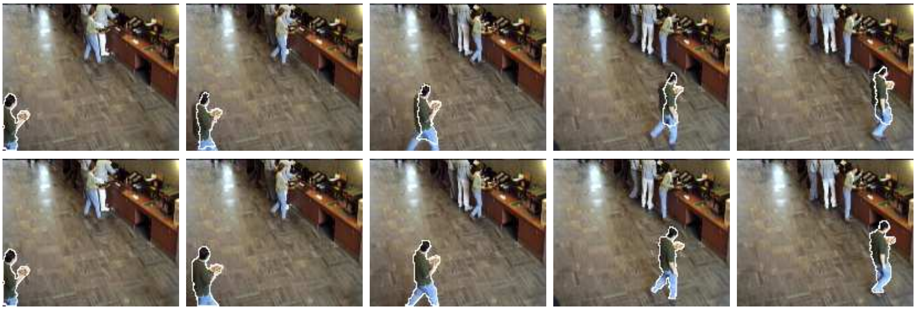


Fig. 2. Example of tracking using foreground matching (first row) and the proposed model (second row). In each of these rows, we show, from left to right, frames 1509, 1510, 1514, 1518 and 1521.

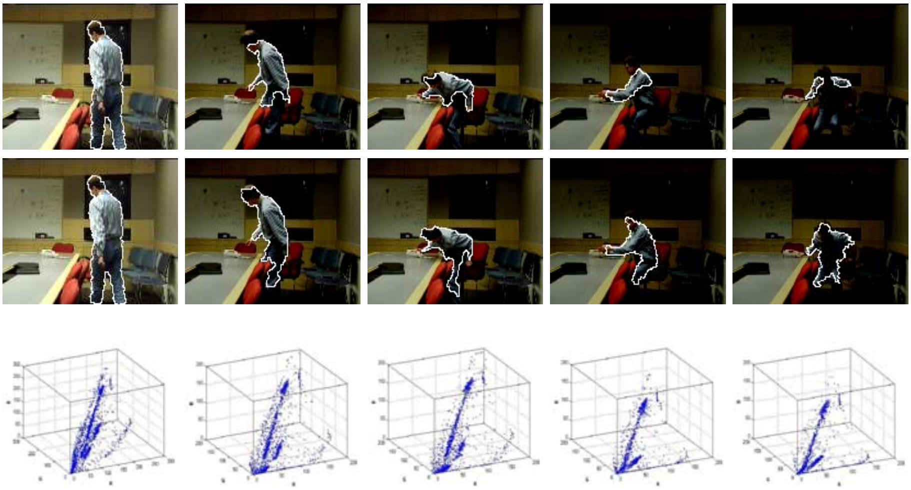


Fig. 3. Example of tracking under illumination change, using foreground matching (first row) and the proposed model (second row). We show in each row, from left to right, frames 1398, 1400, 1404, 1407 and 1488. The last row shows the RGB color distribution of the frames.

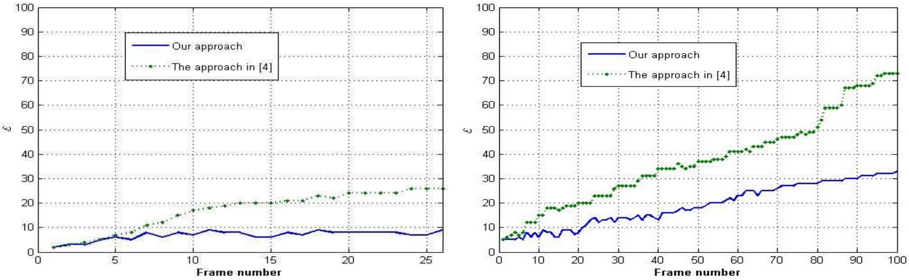


Fig. 4. Quantitative evaluation of the proposed approach. The graphs on the left and the right show, respectively, values of the error $\mathcal{E}_\ell$ for the first and second examples.

where "−" stands for the set difference operator and $|\cdot|$ designates the cardinality of a set. R_o^ℓ and G_o^ℓ designate the tracking result and the ground truth in frame I_ℓ, respectively. Basically, the error $\mathcal{E}_\ell$ gives the percentage of misclassified pixels by the tracking. In another words, it measures the deviation of the zero level set from the real boundaries of the tracked object. Fig. (4) shows the values of $\mathcal{E}$ for the above examples with respect to each of the tested methods. We can see clearly, for both examples, that using *background matching* improves substantially the tracking accuracy.

Finally, we should put a comment on the computation time of our algorithm. We implemented the tracking module using C++ and our tests were run on a Pentium IV 2.4 GH. Currently, our algorithm is able to process two frames per second. Further optimization is in perspective to enhance the rapidity of our approach.

4 Conclusion

We proposed a new model for object tracking by combining foreground and background matching using active contours. The model allows for efficient object tracking under cluttered backgrounds and appearance changes. Our experiments demonstrated these capabilities and enhanced performance compared to foreground matching-based tracking. In the future, we aim to make our approach faster and apply it to specific object tracking (ex. faces, pedestrians, etc.).

References

1. Allili, M.S., Ziou, D.: Object Tracking in Videos Using Adaptive Mixture Models and Active Contours. Neurocomputing 71(10-12), 2001–2011 (2008)
2. Allili, M.S., Ziou, D.: Globally Adaptive Region Information for Automatic Color-Texture Image Segmentation. Pattern Recognition Letters 28(15), 1946–1956 (2007)
3. Avidon, S.: Ensemble Tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 494–501 (2005)
4. Collins, R.T., Liu, Y.: Online Selection of Discriminative Tracking Features. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
5. Bishop, C.M.: Patt. Recog. and Mach. Learn. Springer, Heidelberg (2006)
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Trans. on Patt. Analysis and Mach. Intelligence 25(5), 564–577 (2003)
7. Freedman, D., Zhang, T.: Active Contours for Tracking Distributions. IEEE Trans. on Image Processing 13(4), 518–526 (2004)
8. Nascimento, J.C., et al.: Performance Evaluation of Object Detection Algorithms for Video Surveillance. IEEE Trans. on Multimedia 8(4), 761–774 (2006)
9. Osher, S., et al.: Fronts Propagating With Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. J. of Comp. Physics 79(1), 12–49 (1988)

Processing of Microarray Images

Fernando Mastandrea and Álvaro Pardo

Department of Electrical Engineering - Universidad Católica del Uruguay
fmastand@ucu.edu.uy, apardo@ucu.edu.uy

Abstract. In this paper we present the results of a system for processing microarray images which includes the gridding and spot detection steps. The main goal of this work is to develop automatic methods to process microarray images including confidence measures on the results. The gridding step is based on the method proposed in [1] and improves it by the automatic determination of the grid parameters, and a more precise orientation detection. For spot detection the algorithm uses the Number of False Alarms methodology [2] which can be used to finely adjust the spot position and provides a confidence measure on the detection. We test the results obtained by our method with simulated images against existing microarray software.

1 Introduction

Microarray technology allows comparative experiments on gene expression. Each array consists on thousands of regularly placed spots which contain control and test samples. The samples are labeled with two different green (Cy3) and red (Cy5) fluorescent dyes. After the biological reaction takes place the digital image is obtained using a microarray scanner. The intensity of each pixel indicates the hybridization of the control and test samples. Once the image has been acquired, it is processed to extract features at each spot. After that, statistical processing is used to reveal the gene expression levels.

The analysis of microarray images involves the detection of the features that will later be used to infer the results on the experiment. The statistics that will be gathered for inference will be calculated from the pixel values of the detected spots. Therefore the correct detection of spots is crucial for data extraction.

The processing of microarray images can be divided into three steps: grid detection, spot detection and data extraction [4]. In the first step a template of the grid must be adjusted to the acquired image. In this step it is also useful to automatically learn some parameters of the grid, like spacing between spots, angles, etc. The result of this step provides candidate centers for each spot. Based on this information the detection of each spot is refined. Finally, in the last step, information from each spot is extracted for later analysis.

Although at first glance these problems may seem simple, microarrays may contain noise and distortions that deteriorate the results of these steps: small dots/speckles which can be confused with spots, artifacts contaminating the spot, missing spots (blanks), donut shaped spots and also entire columns (or rows) can be very dim, among others.

In order to assist the technician in the process of microarray analysis we need automatic and semiautomatic methods which provide a confidence measure on the detection

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 962–969, 2009.

results. In order to reach that goal we use results from the Computational Gestalt Theory (see Section 2.2) which gives a confidence measure that can be used by the user.

As mentioned earlier, we based part of the grid detection in [1], but it has some differences. First, the minimum and maximum radii of a typical spot are estimated automatically. Second, the angles which represent the orientation of the grid are found by interpolation, giving better accuracy without being too computationally expensive. Third, we don't use a regular grid model nor MRF to refine spot center coordinates. However, our gridding method allows having different distances between rows and columns of spots and the spot center coordinates are refined by the segmentation algorithm. We also obtain the grid coordinates with subpixel accuracy. As for the segmentation, we added a method for spot detection which includes confidence estimation on the detection (see Section 2.2).

2 Proposed Method

2.1 Gridding

The gridding step consists in finding the approximate spot center coordinates. This information will be used later in the segmentation step. Usually, a microarray is composed of several grids arranged in matrix form in the image. The subgrids follow the same layout of spot rows and columns. Since the layout parameters are known beforehand here we concentrate on the extraction of spots on subgrids. The proposed algorithm has only two parameters: the number of rows and columns of the grid, sr and sc respectively. In Fig.5 we show the GUI of the developed software.

Radius estimation. In this step we estimate the mean spot radius using a heuristic procedure. For now on if the image is bi-channel, the average intensity image I is used (see Fig. 1). First we apply histogram equalization and stretching to I (see Fig. 1(b)). Next, we compute k different thresholds so that $t_i = \frac{i}{k} \cdot (I_h - I_l) + I_l$ with $i = 1 \ldots k$. Where I_h and I_l correspond to the maximum and minimum of I. Now for every iteration i, we apply the threshold t_i to the image obtaining U_t. We show in Fig. 1(c) for a threshold t_i with $i = k/2$. For every U_t we apply the following algorithm:

```
1: Remove isolated pixels and fill holes in $U$, store the result in $R$.
2: Remove all regions of $R$ which area is less than $\pi * (R_{min})^2$.
3: Label remaining regions of $R$.
4: **if** there is more than one region left **then**
```

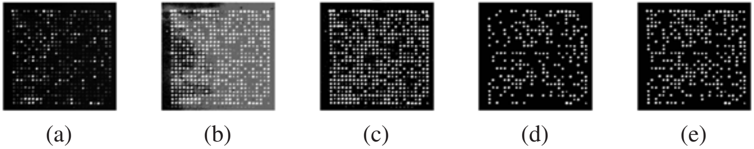


Fig. 1. (a) Input grid image. (b) Grid image followed by histogram equalization and stretching. (c) Thresholded grid image. (d) Regions passing constraints. (e) Final binary image.

```
    Discard regions based on compactness[1] ($0.75 < c < 1.25$) and eccentricity[2] ($0 < e < 0.5$).
    Generate a binary image $B$ from all remaining regions.
    Perform bitwise OR with I, so that: $I = I|B$
end if
```

After the k iterations the result is a binary image which contains most spots (see Fig. 1(e)). Finally we take all regions and calculate their median area $\bar{a}$ and estimate the radius as $\hat{r} = \sqrt{\frac{\bar{a}}{\pi}}$. The parameter k is the threshold granularity, we use $k = 10$. R_{min} is used for filtering out artifacts and a value of 2 pixels seemed satisfactory in our experiments.

Angle estimation. The first step in the gridding process is to identify the angles α and β that determine the orientation of the grid columns (sc) and rows (sr) of spots. This step is partly based in the procedure described in [1]. The following paragraphs are a brief explanation of the the orientation estimation procedure presented in [1].

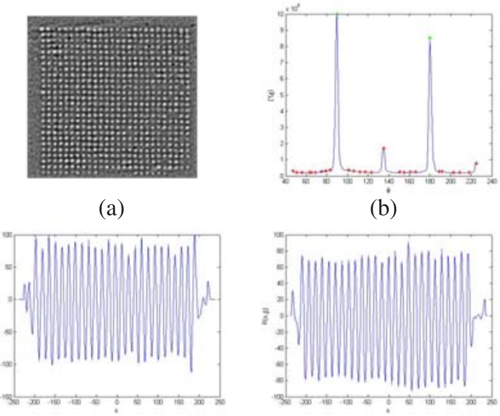
Initially we apply the Orientation Matching (OM) transform to the grid image $I(x, y)$ obtaining $OM\{I\}(x, y)$. Given the percentage v of spot radius variability we define the following maximum and minimum radii for the OM transform: $R_M = r(1 + v/100)$ and $R_m = r(1 - v/100)$). The OM transform provides us with an image intensity values in the range $[-1, 1]$ which represent the match between the gradient of the image and the normals of an annulus of R_m and R_M radii centered on (x, y). The OM transform of the grid image of Fig.1 is shown on Fig.2. We filter the OM image with a median filter of size $[5 \times 5]$ since our experiments had shown the following steps benefit from an image with less noise. Next, we apply the Radon Transform (RT), obtaining $\mathcal{R}\{OM(I)\}(s, \phi)$. We then integrate the s variable obtaining $\Gamma(\phi) = \int_s \mathcal{R}^2\{OM(I)\}(s, \phi)\, ds$. Then we low-pass filter $\Gamma(\phi)$ (with the same parameters as specified in [1]) and take the two maximum values $m_1 = \Gamma(\phi_a)$ and $m_2 = \Gamma(\phi_b)$ corresponding to the principal orientations ϕ_a and ϕ_b (see Fig. 2(b)). A typical grid with little rotation will have it's maxima around $\phi \approx 90$ and $\phi \approx 180$. So we choose ϕ_a as the radon angle closer to $90°$, and ϕ_b closer to $180°$. Up to this point the method is the same as in [1].

Because the maxima are expected to be at $90°$ and $180°$, we calculate ϕ_i ranging from $45°$ to $225°$ instead of from $0°$ to $180°$. So ϕ is the vector $\phi = \{\phi_0, \phi_i, \ldots, \phi_n\}$, where $\phi_0 = 45$ and $\phi_n = 225$. We do that in order to avoid having a maximum of $\Gamma(\phi)$ which would correspond to the grid orientation at the end point. It follows the higher is n the more precision we have for $\Gamma(\phi)$. However, since the RT has to be calculated with a given angle step, increasing n to obtain more precise angles is more computationally demanding. As an alternative we can propose interpolation.

Angle interpolation. Given ϕ_a and ϕ_b we improve angle estimation via interpolation. We find a new interpolated angle α by taking the points $(\phi_{a-1}, \Gamma(\phi_{a-1}))$, $(\phi_a, \Gamma(\phi_a))$ and $(\phi_{a+1}, \Gamma(\phi_{a+1}))$ and solve the parabola $y = ax^2 + bx + c$ that passes through these

[1] $c = \frac{P^2}{4\pi A}$.

[2] Defined as the ratio of the distance between the foci of the ellipse and its major axis length. $0 \leq e < 1$.



(a) (b)

(c) (d)

Fig. 2. (a) OM transformed image of the grid. (b) $\Gamma(\phi)$ graph. The peaks correspond to the angles ϕ_a and ϕ_b. Profiles (c):$\alpha = 89.616$ and (d):$\beta = 180.15$.

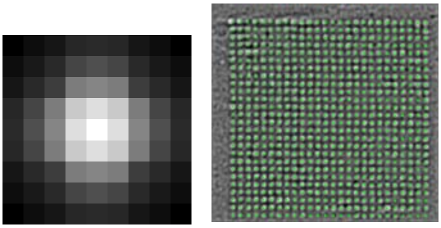
points. We then take the maximum value as the new $\alpha = -b/(2a)$. In the same way the anlge β is computed.

Obtaining grid rows and columns. Given the new interpolated angles, we find the profiles given by rotating (using interpolation) the grid image with α and β. This step provides us of two profiles, which are the radon profiles $\mathcal{R}\{OM(I)\}(s,\alpha)$ and $\mathcal{R}\{OM(I)\}(s,\beta)$ (see Fig. 2).

To simplify the notation we use $\mathcal{R}(s,\phi)$ instead of $\mathcal{R}\{OM(I)\}(s,\phi)$. Now, for every s, $\mathcal{R}(s,\alpha)$ represents the OM image projected with the direction α. The angle α is such that the local maxima of $\mathcal{R}(s,\alpha)$ indicate the spacing between the rows of the grid, as seen in Fig.2. Lets call the radon distances where the local maxima occur $\{s^\alpha\} = \{s_1^\alpha, s_2^\alpha, \ldots, s_{n_1}^\alpha\}$. In a similar way, the maxima of $\mathcal{R}(s,\beta)$ indicate the spacing of the columns, and we call it $\{s^\beta\} = \{s_1^\beta, s_2^\beta, \ldots, s_{n_2}^\beta\}$. So, we consider normal lines originating at the center of the image, with angle α and distance $\{s^\alpha\}$ and call them $\{l^\alpha\}$. We do the same with angle β and distance $\{s^\beta\}$, getting $\{l^\beta\}$. We are using the peaks in the profile as indicators of the location of the spots rows and columns. We are going to use this information to construct a non-regular grid (not the same distance between spot rows and columns).

Up to this point we have the profiles in function of s. Using the same ideas as in **Angle interpolation** we find the maximum values of the profiles. The process is the same, but instead of interpolating the angles vector we apply it on the distances vector s. We obtain a set of interpolated distances which we call $\{s^\alpha\}$ and $\{s^\beta\}$. These distances, in subpixel resolution, represent the spot rows and columns plus noise, as we will see next.

Filtering erroneous maxima. If the array image was ideal, we would expect to find the spot centers in the intersection of the two set of lines $\{C\} = \{l_\alpha\} \cap \{l_\beta\}$. Or equivalently, we would expect for $\{s^\alpha\}$ and $\{s^\beta\}$ to have $n_1 = sr$ and $n_2 = sc$ elements



(a) (b)

Fig. 3. (a) Spot template. (b) The output of the gridding method.

respectively. But, in a real grid image, noise takes part in erroneous detection of the maxima of $\mathcal{R}(s, \alpha)$ and $\mathcal{R}(s, \beta)$ as can be seen in Fig. 2. Note that, in the first profile we have 29 maxima and $sr = 26$. To filter out these false maxima we propose the following iterative procedure to remove elements in the set of distances $\{s^\alpha\}$ and $\{s^\beta\}$. We iterate through the distance differences vector d, and if some element falls below the threshold T, the algorithm removes the element of $\{s\}$ so that the new $\{s\}$ has differences closer to its median. In other words this filter tries to remove false spot rows or columns which are in between the real rows and columns. After this process we have the set of distances $\{s^\alpha\}$ and $\{s^\beta\}$ with some erroneous elements removed and tentative spot center coordinates at the intersection of corresponding lines.

Grid placement. At this point we have a set of spot centers but there can still be erroneous centers due to noise. In this step we build a grid, based on the known number of spot columns and rows and the coordinates we already have, to find the best match of that generated grid to the image. We start by making a spot template, see Fig. 3, as a disc with radius r obtained earlier. We OM transform this disc since the input image for this step is the OM transformed grid image. Then we generate a grid with $sr \times sc$ deltas centered at the intersection of previously found lines. Since wrong line detection generates false spot centers, we could have detected more spot centers than the ones present in image. To select the correct spot centers we generate several grids of deltas with the known number of rows and columns based on the number of spot center coordinates we found previously. Then we convolve each delta grid and the spot template to obtain a grid template. Finally, we find the correlation of the template grids with the OM transformed grid images, and select the template grid that best matches the image. After the steps presented above we have a grid that best matches the input image, its parameters and the spots parameters (center and radius). Next, we present a segmentation method to refine the spot center coordinates.

2.2 Spot Segmentation Using Computational Gestalt Theory

Computational Gestalt Theory was first presented by Desolneux, Moisan and Morel [2] as a way to obtain a quantitative theory of the Gestalt laws. Computational Gestalt uses the *Helmholtz Principle* to define a quantitative measure of a given gestalt [2].

Helmholtz Principle. The observation of a given configuration of objects in an image is meaningful if the probability of its occurrence by chance is very small. The Helmholtz principle can be formalized by the definition of the *Number of False Alarms* and ϵ-*meaningful events:*

Number of false alarms - NFA. The number of false alarms (NFA) of an event E is defined as: $\mathrm{NFA}(E) = \mathcal{N} \cdot \mathrm{P}[\mathcal{E} \geq E|H_1]$ where $\mathcal{N}$ is the number of possible configurations of the event E and H_1 is the background or *a contrario* model. An event E is ϵ-meaningful if the NFA is less than ϵ: $\mathrm{NFA}(E) < \epsilon$.

Spot Segmentation. Using the center coordinates and radius estimated before we apply a threshold based segmentation. The optimal threshold which separates the spot from the background is estimated with an NFA approach.

This method is applied to a small, square image centered on the spot of size $2r + 1$. Given a threshold, t, we compute the number of pixels, k_o, outside the spot with grey level above t. If N_o is the total number of pixels outside the spots and p_s is the probability of a pixel being above the threshold t[3] we can estimate the probability of at least k_o pixels above the threshold among N_o using the binomial distribution. Therefore, in this case the NFA is computed using the binomial tail as: $N_T \times B(p_s, n_o, k_o)$ where N_T is the number of thresholds tested. Additionally, with this procedure the NFA is a confidence measure which tells us if there is a spot or not in this position. Spots with $NFA > \epsilon$ are not considered in following steps. We iterate this procedure in a small region around the spot center given by the gridding step. We settle with the coordinates that give the best NFA figure. Therefore obtaining a better estimate of the spot center coordinates.

Data extraction. Our software also includes this step. Due to lack of space we do not present or evaluate this step here.

3 Results

In this section we show a comparison between the results obtained by our method and the program UCSF spot[4]. We used simulated microarray images with noise and distortions generated by *mamodel* [3].

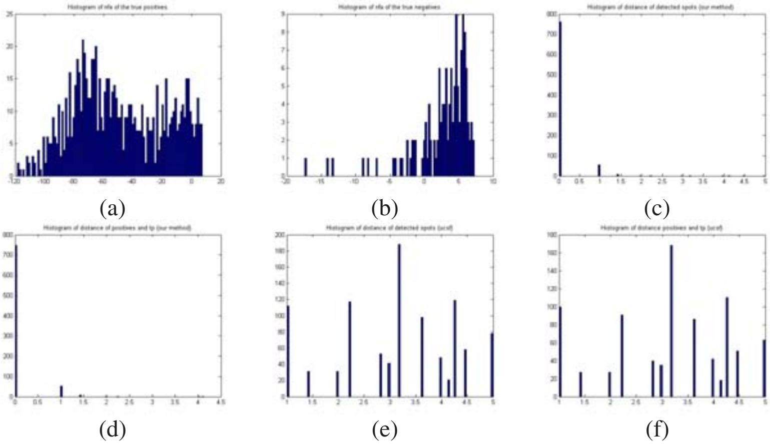
In the *mamodel* website there are three parameter sets to generate different images. Their description read: “High quality slide”, “Noisy slide” and “Disturbing noise”. We chose the last one and made the necessary adjustments to the parameters for generating one grid of size 40x25 spots providing us a grid of 1000 spots.

As can be seen in Fig. 5 each image channel is generated with the same spot intensities and some distortions are present in both channels (scratches, air bubbles). However, stains can be present in one or both channels.

We begin our tests by obtaining the true positives (TP), that is, spots that should be found. mamodel provides in its output the noise free intensity values for each spot. In a posterior process, mamodel generates the noise contaminated slide image from those values. So, a reasonable approach would be to flag a spot as negative if its original intensity falls below the background noise level. For this matter, we took a region of the

[3] The probability p_s is empirically estimated based on the values of the pixels of a square region around the spot.

[4] www.jainlab.org



(a) (b) (c)

(d) (e) (f)

Fig. 4. (a) NFA of TP. (b) NFA of true negatives. (c-f) Histogram of distances from the reference center coordinates with: (c) Our method for TP, (d) Our method for all positives, (e) UCSF for positives, (f) UCSF for positives.

image containing only background noise and found its mean and obtain the true positives, as explained above. Although simple, this procedure has one obvious drawback: we are not taking into account any distortion, for example scratches, air bubbles, etc.

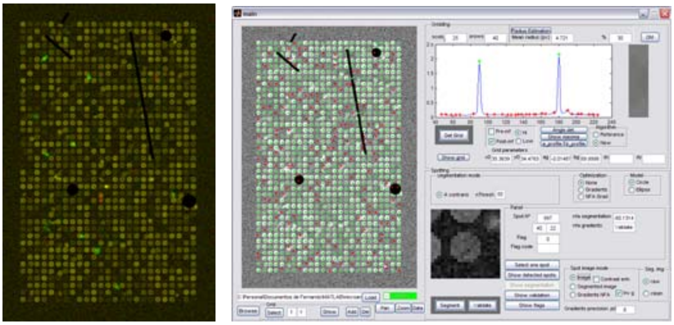
In Fig.4(a) and (b) we show the histogram of the NFA values for the TP and the true negatives (TN). Note that the NFA value can be used to flag a spot as found or missing as correctly discriminates between TP and TN. Also note that in Fig.4(a) there are still a significant amount of spots with NFA>0. Manual inspection of those spots confirmed that is caused by our imperfect way of flagging a spot as TP, without taking into account the distortions in the image as mentioned earlier.

If we consider a spot as positive if its NFA<0 we obtained the results in the following table with false positives (FP) and false negatives (FN). As we can see our method produces a more balanced pair sensitivity-specificity.

Value	Our method	UCSF Spot
Number of FN	50	5
Number of FP	22	137
Sensitivity	94.5%	99.4%
Specificity	86.2%	50%

Now we compare the accuracy on spot center detection for our method and UCSF. We take into account only the spots flagged as found by each program. We computed the distances to the reference spot center coordinates given by mamodel and plotted its histogram shown in Fig.4. As we can see for TP our method gives zero error for most of the spots while UCSF has errors ranging from 1 to 5. Regarding all detected spots (positives) our method gets some spots with larger errors but still the great majority has error cero as can be seen in the histograms. The statistics are presented in the following table.

Value	Our method (positives)	Our method (positives and TP)	UCSF Spot (positives)	UCSF Spot (positives and TP)
Mean	0.1267	0.0980	3.1488	3.1552
Median	0	0	3.1623	3.1623
Variance	0.2501	0.1456	1.3497	1.3554



(a) (b)

Fig. 5. (a) Simulated Image. (b) Screen of the developed software prototype showing the detected spots: green $NFA \leq 0$, red $NFA > 0$.

4 Conclusion

We developed a software prototype for the analysis of microarray images which includes the stages of gridding, segmentation and data extraction (not presented here). Starting from the method proposed in [2] we introduced several improvements to increase the accuracy. For the segmentation step we presented a method based on NFA which provides a confidence measure that can be used to flag spot and assist the user during manual inspection. We compared the results of our method with UCSF and outperformed it in sensitivity-specificity and spot center detection.

References

1. Ceccarelli, M., Antoniol, G.: A deformable grid-matching approach for microarray images. IEEE Transactions on Image Processing 15(10), 3178–3188 (2006)
2. Desolneux, A., Moisan, L., Morel, J.-M.: From Gestalt Theory to Image Analysis: A Probabilistic Approach. Springer, Heidelberg (2008)
3. Nykter, M., Aho, T., Ahdesmaki, M., Ruusuvuori, P., Lehmussola, A., Yli-Harja, O.: Simulation of microarray data with realistic characteristics. BMC Bioinformatics 7, 349 (2006)
4. Yang, Y.H., Buckley, M.J., Speed, T.P.: Analysis of cdna microarray images. Briefings in Bioinformatics 2(4), 341–349 (2001)

Multi-focus Image Fusion Based on Fuzzy and Wavelet Transform

Jamal Saeedi[1], Karim Faez[1], and Saeed Mozaffari[2]

[1] Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
{jamal_saeidi,kfaez}@aut.ac.ir
[2] Semnan University, Electrical and Computer Department, Semnan, Iran
mozaffari@semnan.ac.ir

Abstract. In this paper, we proposed a new method for spatially registered multi-focus images fusion. Image fusion based on wavelet transform is the most commonly fusion method, which fuses the source images information in wavelet domain according to some fusion rules. There are some disadvantages in Discrete Wavelet Transform, such as shift variance and poor directionality. Also, because of the uncertainty about the source images contributions to the fused image, designing a good fusion rule to integrate as much information as possible into the fused image becomes one of the most important problem. In order to solve these problems, we proposed a fusion method based on double-density dual-tree discrete wavelet transform, which is approximately shift invariant and has more sub-bands per scale for finer frequency decomposition, and fuzzy inference system for fusing wavelet coefficients. This new method provides improved subjective and objectives results compared to the previous wavelet fusion methods.

Keywords: Image fusion, double-density dual-tree discrete wavelet transform, fuzzy classifier, multi-focus.

1 Introduction

Image fusion provides a means to integrate multiple images into a composite image, which is more appropriate for the purposes of human visual perception and computer-processing tasks such as segmentation, feature extraction and target recognition. Important applications of the fusion of images include medical imaging [1], microscopic imaging, remote sensing [2], computer vision, and robotics [3].

Fusion techniques include the simplest method of pixel averaging to more complicated methods such as principal component analysis [4], and multi-resolution fusion [5]. Multi-resolution images fusion is a biologically-inspired method, which fuses images at different spatial resolutions. Similar to the human visual system, this fusion approach operates by decomposing the input images into a resolution pyramid of numerous levels. Each level contains one or more bands representing orientation or detail/approximation information. Following this decomposition, the fusion now takes place between the corresponding coefficients or samples in each band. The fused pyramid is then reconstructed to form the final fused output image.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 970–977, 2009.

Nick Kingsbury has introduced DT-CWT [6], which introduces limited redundancy (4X) and allows the transform providing approximate shift invariance and directionally selective filters while preserving the usual properties of perfect reconstruction and computational efficiency. There are many publications, which used DT-CWT for fusion schemes and showed better subjective and objective results [7]. In this paper we proposed a new algorithm based on double-density dual-tree DWT [8], which is an over complete discrete wavelet transform (DWT) designed to simultaneously possess the properties of the double-density DWT [9], and the dual-tree complex DWT [6].

The three previously important developed fusion methods, which were implemented in wavelet transform domain, are as follows: Maximum selection (MS), which just picks the coefficients in each sub-band with the largest magnitude; Weighted average (WA), which is developed by Burt and Kolczynski [10] and used a normalized correlation between the two images sub-bands over a small local area. The resultant coefficients for reconstruction are calculated from this measure via a weighted average of the two images coefficients; Window based verification (WBV), which is developed by Li et al. [11] and creates a binary decision map to choose between each pair of coefficients using a majority filter.

These fusion rules ignore some useful information and are sensitive to noise. Selective operation made the fused coefficients completely dependent on the coefficients with larger average of local area energy and ignores the other corresponding coefficient. In the weighted average scheme, the weights were computed by a linear function which cannot describe the uncertainty of each source image contributions. Also in coarser level of decomposition, fusion task is much harder and these fusion rules do not work very well. In order to solve these uncertainties and information integration, this paper proposed a new fusion algorithm, which also is based on new wavelet transform and new fusion rule based on fuzzy classifier.

The paper is structured as follows: In Section 2 we describe briefly about the double density dual tree DWT. In section 3 the proposed fuzzy image fusion is explained. Section 4 gives various results and comparisons. Finally, we conclude with a brief summary in section 5.

2 Double-Density Dual-Tree DWT

A double-density dual-tree DWT [8] is proposed by Selesnick in 2004. Important refinements in DD-DT-DWT provide filters that are nearly shift-invariant with vanishing moments, compact support, and a high degree of smoothness.

The 2-D DD-DT-DWT has a total of 32 oriented real wavelets or 16 complex wavelet sub-images per level, while the DT-DWT has 12 oriented real wavelets or 6 complex wavelet sub-bands filters per level. This structure is sometimes described by a parent children- grandchildren genealogy (e.g. parents start at level 3, children at level 2, and grandchildren at level 1). The DD-DT-DWT by comparison with the DT-DWT can has wavelets, which are more closely spaced spatially, or wavelet sub-bands which are more closely spaced with respect to scale. It also has more sub-bands per scale for finer frequency decomposition with increased wavelet regularity for same length filters. However, 2-D DD-DT-DWT requires 10.66X rather than 4X memory increase in DT-DWT. Nevertheless, because of their finer regular sub band coverage, the DD-DT-DWT will be used in this research.

3 Proposed Fuzzy Image Fusion

In this scheme, the fusion output achieved by combination of three inputs obtained with three different fusion rules. In fact the fuzzy system specifies three inputs contribution in the final output. These fusion rules can be explained as follows:

3.1 Fusion Using Decision Map

This rule forms the first input of the fuzzy system using a logical matrix, called decision map. In many publications local features is used to generate the decision map, for selecting coefficients between high frequency sub-bands of two images such as mean and standard deviation [12], energy, gradient, fractal dimension, contrast, and standard deviation [2], spatial frequency, visibility, and information entropy [13], for image fusion. Here, we used a combination of two features for generating confident decision map.

The high frequency coefficients reflect the image edge and detail information. According to imaging mechanism of optical system, the bandwidth of system function for images in focus is wider than that for images out of focus. Therefore the pixel values of clear images are larger than that of blurred images.

In order to enhance this information, we used two texture features. The first feature calculates local range and second one calculates local energy of the high-frequency sub-bands. We calculate the two features using:

$$Range\colon F_1^{\,k}(x,y) = \max_{x,y\in W}\left|sb_{k_i}^{\;j}(x,y)\right| - \min_{x,y\in W}\left|sb_{k_i}^{\;j}(x,y)\right| \tag{1}$$

$$Energy\colon F_2^{\,k}(x,y) = \sum_{x,y\in W}\left(sb_{k_i}^{\;j}(x,y)\right)^2 \tag{2}$$

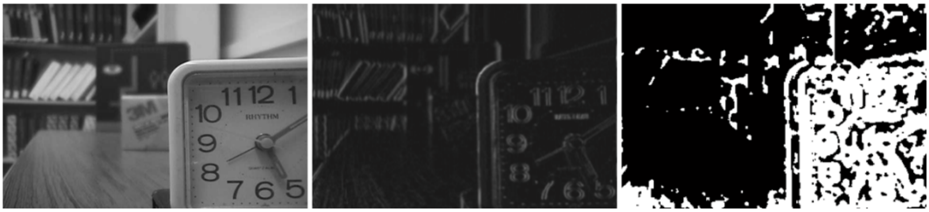
where $j = 1,2 \ldots N-1$, which N is the level of the decomposition, $i = 1,2 \ldots 16$, which denote the sixteen sub-bands of high frequency coefficients at each level, $k = 1,2$, which is the number of images, and W is the local window.

Also for improving these features a nonlinear averaging filter is used for reducing noise and taking into accounts neighbor dependency. This operation implement as follows:

$$NF_i^{\,k}(x,y) = \frac{\sum_{a=-A}^{A}\sum_{b=-B}^{B}\mu(a,b)\times F_i^{\,k}(x,y)}{\sum_{a=-A}^{A}\sum_{b=-B}^{B}\mu(a,b)} \tag{3}$$

where $i = 1,2$, which is two texture features, $k = 1,2$ is the number of images and $[2\times A+1, 2\times B+1]$ is the size of local window. Also $\mu(a,b)$ is calculated using:

$$\mu(a,b) = \exp\left[-\left(\frac{a^2+b^2}{NW}\right)\right] \tag{4}$$



(a) (b) (c)

Fig. 1. (a) Right-focus “Disk” image. (b) 2th Sub-band at first level of decomposition for right focus image. (c) Decision map for 2th sub-band.

where NW is the number of pixel in the local window and $a \in \{-A \dots A\}, b \in \{-B \dots B\}$. Having the two features, decision map is calculated using:

$$dm_i^j = \begin{cases} 1 & if \quad NF_1^1 > NF_1^2 \; and \quad NF_2^1 > NF_2^2 \\ 0 & \qquad otherwise \end{cases} \tag{5}$$

For example a decision map is obtained for the “Disk” images, which shows in the Figure 1. Finally the first rule output is calculated using:

$$Y_1 = dm_i^j \times sb1_i^j + \left(1 - dm_i^j\right) \times sb2_i^j \tag{6}$$

3.2 Fusion Using Finer Level Decision Map

Most of the fusion rules for merging wavelet coefficients [10], [11] do not work well in coarser level of decomposition. This is happened because of in the coarser level of decomposition there is not sufficient different between features for generation of desired decision map. Therefore we used an estimation of finer level decision map via down-sampling or interpolation for fusing sub-band of two images in coarser levels. Figure 2 shows estimation of decision map for coarser level. The output of second rule can be defined by following equation:

$$Y_2 = dm_i^{j-1} \times sb1_i^j + \left(1 - dm_i^{j-1}\right) \times sb2_i^j \tag{7}$$

where $j = 2,3 \dots N - 1$, which N is the level of the decomposition, $i = 1,2 \dots 16$, which denote the sixteen sub-bands of high frequency coefficients at each level. For $j = 1, Y_2$ obtained via first rule.

Spatial correlation between wavelet coefficients in different levels, which is called *inter-scale dependency* is the idea behind this fusion rule, which is used in many publication for wavelet based compression and denoising [14].

3.3 Fusion Using Averaging

In the smooth region of two images that may be existed in focus or out of focus region in the image we cannot take a good decision for fusion task, because there is not sufficient difference between them for distinguishing in and out of focus regions. Therefore we used a simple fusion rule, which is averaging that can remove Gaussian noise. This rule can be defined using:

$$Y_3 = \frac{sb_{1i}^{\ j} + sb_{2i}^{\ j}}{2} \tag{8}$$

3.4 Fuzzy Classifier

We want to design a good fusion rule with combining these three fusion rules to integrate as much information as possible into the fused image. We used a fuzzy classifier for this purpose.

Here we used fuzzy rule-based classifier. The simplest fuzzy rule-based classifier is a fuzzy if-then system, similar to that used in fuzzy control [15]. We labeled output of each fusion rules as a class. This classifier can be constructed by specifying classification rules as linguistic rules:

1. IF NF_1 is *large* AND NF_2 is *large* THEN Y_1 is output.
2. IF NF_1 is *large* AND NF_2 is *small* THEN Y_2 is output.
3. IF NF_1 is *medium* AND NF_2 is *large* THEN Y_1 is output.
4. IF NF_1 is *medium* AND NF_2 is *small* THEN Y_2 is output.
5. IF NF_1 is *small* THEN Y_3 is output.

where $NF_1 = |NF_1^1 - NF_1^2|$, and $NF_2 = |NF_2^1 - NF_2^2|$.

Each linguistic value is represented by a membership function. Figure 3 shows triangular membership functions for NF_1, which is normalized and T_1 is a constant value. For the pair of values (NF_1, NF_2), the degree of satisfaction of the antecedent part of the rule determines the firing strength of the rule. The firing strengths of these rules are calculated as:

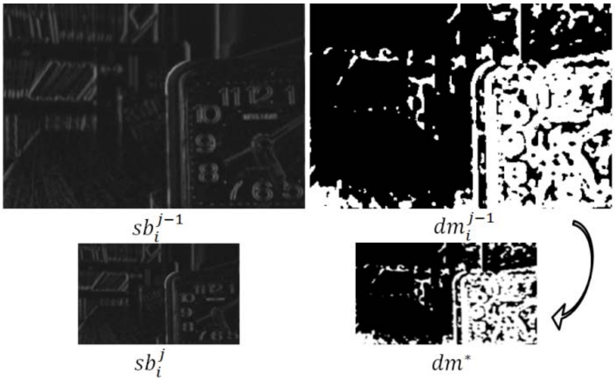


Fig. 2. Estimation of decision map $(\boldsymbol{dm}^*)$ for coarser level

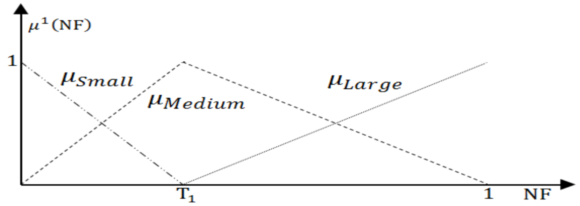


Fig. 3. Fuzzy membership function for the linguistic terms of $\boldsymbol{NF_1}$

$$\tau_1 = \mu^1_{large}(NF_1) \times \mu^2_{large}(NF_2), \tau_2 = \mu^1_{large}(NF_1) \times \mu^2_{small}(NF_2)$$

$$\tau_3 = \mu^1_{medium}(NF_1) \times \mu^2_{large}(NF_2), \tau_4 = \mu^1_{medium}(NF_1) \times \mu^2_{small}(NF_2)\ \tau_5 = \mu^1_{small}(NF_1)$$

The AND operation is typically implemented as minimum but any other t-norm may be used. We have chosen algebraic product for the AND operation.

The rules "vote" for the class of the consequent part. The weight of this vote is τ_i. To find the output of the classifier, the votes of all rules are aggregated. Among the variety of methods that can be applied for this aggregation, we considered the maximum aggregation method. Let k is the class labels, j denote number of rules and $i \to k$ denote that rule i votes for $\mathrm{Y_k}$. Then:

$$If \quad \tau_i = \max_{j=1...5} \tau_j \quad AND \quad i \to k \quad THEN \quad Class \quad is \quad Y_k \tag{9}$$

For building fuzzy membership function T_1 must be defined. We obtained $0.1 \le T_1 \le 0.2$ using test images and try and error. Finally the new sub-band for generating output image is obtained using:

$$sb\_new_i^j = Y_k \tag{10}$$

where $j = 1,2 \dots N - 1$, which N is the level of the decomposition, $i = 1,2 \dots 16$, which denote the sixteen sub-bands of high frequency coefficients at each level.

Also fusion rule for low-frequency sub-bands is defined by:

$$sb\_new_i^N = \frac{sb_{1i}^{\ N} + sb_{2i}^{\ N}}{2} \tag{11}$$

where $i = 1,2$, which is low frequency sub-bands in the last level. After merging the wavelet coefficients, the final fusion result is obtained by inverse wavelet transform.

4 Experimental Results

The images used in these experiments are selected from multi-focus datasets; publicly available at the Image fusion web site [18] (Figures 4). To compare our image fusion method, the image fusions based on the DWT [5], shift invariant DWT (SIDWT) [16], and DT-CWT decompositions [8] are also implemented.

To evaluate our comparisons objectively, the same fusion rules (MS, WA [10], WBV [11], and our proposed method) are used in the DWT, SIDWT, DT-CWT and DD-DT-DWT schemes. The image PSNR and Petrovic index [17], used to evaluate

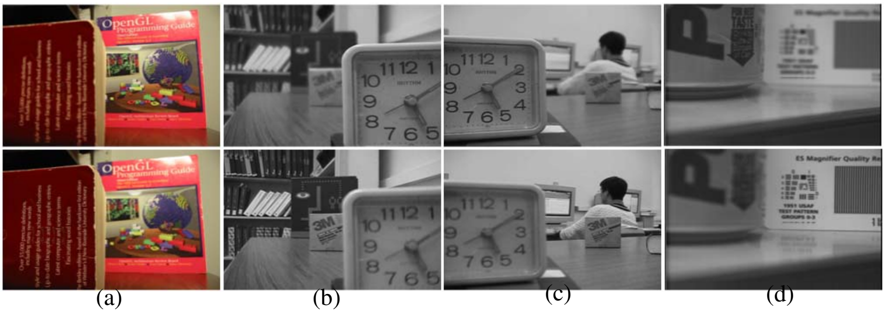

(a) (b) (c) (d)

Fig. 4. Test images used in the experiments. (a)-(d) Book, Disk, Lab and Pepsi, respectively.

the fused image. It should be mentioned that for image fusion experiment, a ground-truth image was used by cutting and pasting method. Subjective results show better visual effect without any artifact compared to other fusion schemes. Notice to the artifacts around the head in the Figure 5 (a)-(c) images. Also objective results in the Tables 1 obviously indicate that our fusion scheme is better than others.

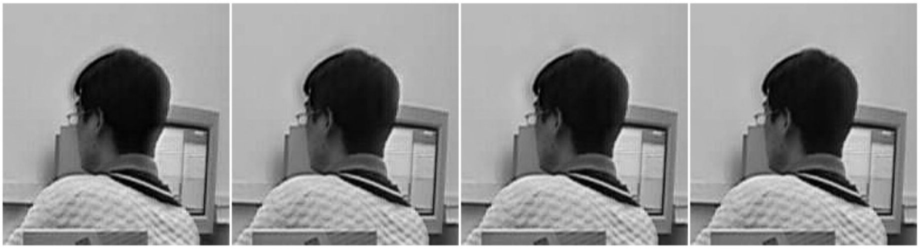

(a) (b) (c) (d)

Fig. 5. Subjective fusion results, Fusion result of a part of "Lab" image using DD-DT-CWT and (c) MS (d) WA (e) WBV (f) proposed method

Table 1. objective image fusion results

		"Disk" 640 × 480		"Lab" 640 × 480		"Pepsi" 512 × 512		"Book" 1024 × 768	
Method		**PSNR**	**Petrovic**	**PSNR**	**Petrovic**	**PSNR**	**Petrovic**	**PSNR**	**Petrovic**
DWT(Haar)	MS	32.6	0.647	34.2	0.644	38.3	0.734	34.3	0.634
	WA	36.5	0.681	35.2	0.680	40.3	0.743	37.2	0.668
	WBV	**36.8**	**0.689**	35.7	**0.686**	**40.8**	**0.747**	37.5	0.671
	Fuzzy	35.1	0.682	**35.8**	0.682	40.0	0.754	**38.6**	**0.678**
SIWT(Haar)	MS	35.7	0.672	36.0	0.669	39.4	0.758	35.3	0.644
	WA	36.3	0.683	36.2	0.681	39.6	0.761	37.8	0.678
	WBV	36.7	0.687	36.5	0.685	39.8	0.761	38.1	0.689
	Fuzzy	**36.9**	**0.691**	**37.0**	**0.691**	**40.0**	**0.766**	**39.5**	**0.699**
DT-DWT	MS	35.2	0.667	35.8	0.666	39.2	0.756	36.9	0.684
	WA	36.6	0.690	36.7	0.688	39.8	0.763	39.3	0.698
	WBV	37.1	0.692	37.0	0.692	39.9	0.763	38.6	0.700
	Fuzzy	**38.7**	**0.699**	**38.8**	**0.704**	**41.2**	**0.772**	**41.5**	**0.710**
DD-DT-DWT	MS	34.2	0.678	35.4	0.669	38.9	0.765	37.1	0.691
	WA	36.9	0.697	37.1	0.693	40.3	0.771	39.7	0.699
	WBV	37.3	0.700	38.2	0.699	41.0	0.772	39.3	0.706
	Fuzzy	**39.6**	**0.703**	**40.1**	**0.709**	**41.9**	**0.779**	**42.3**	**0.719**

5 Conclusion

In this paper, we have presented a new multi-focus image fusion method using double-density dual-tree DWT and fuzzy classifier. This new method used DD-DT-DWT for finer frequency decomposition and shift invariant property compared to other wavelet decomposition and fuzzy classifier for fusing sub-bands of two images, because of overcoming uncertainties in other fusion algorithm mentioned before. The experimental results demonstrated that the proposed method outperforms the standard wavelet fusion methods in the fusion of multi-focus images.

References

1. Garg, S., Kiran, U., Mohan, K., Tiwary, R.: Multilevel Medical Image Fusion using Segmented Image by Level Set Evolution with Region Competition. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society, January 17–18, pp. 7680–7683 (2006)
2. Yang, X.-H., Jing, Z.-L., Liu, G., Hua, L.Z.: Fusion of multi-spectral and panchromatic images using fuzzy rule. Communications in Nonlinear Science and Numerical Simulation 12, 1334–1350 (2007)
3. Kam, M., Zhu, X., Kalata, P.: Sensor fusion for mobile robot navigation. Proceedings of the IEEE 85, 108–119 (1997)
4. Kumar, S., Senthil, M.S.: PCA-based image fusion. In: Proceedings of the SPIE, vol. 6233, p. 62331T (2006)
5. Ke, R.Z., Li, Y.-J.: An Image Fusion Algorithm Using Wavelet Transform. Acta Electronica Sinica 32(5), 750–775 (2004)
6. Kingsbury, N.: A Dual-Tree Complex Wavelet Transform with Improved Orthogonality and Symmetry Properties. In: ICIP, vol. 2, pp. 375–378 (2000)
7. Wei, S., Ke, W.: A Multi-Focus Image Fusion Algorithm with DT-CWT. In: International Conference on Computational Intelligence and Security, pp. 147–151 (2007)
8. Selesnick, I.W.: The Double-Density Dual-Tree DWT. IEEE Trans. on Signal Processing 52(5), 1304–1314 (2004)
9. Petrosian, Meyer, F.G.: The double density DWT. In: Wavelets in Signal and Image Analysis: From Theory to Practice, Kluwer, Boston (2001)
10. Burt, P.J., Kolczynski, R.J.: Enhanced image capture through fusion. In: Proceedings of the 4th International Conference on Computer Vision, pp. 173–182 (1993)
11. Li, H., Manjunath, B.S., Mitra, S.K.: Multi-sensor image fusion using the wavelet transform. Graphical Models and Image Processing 57(3), 235–245 (1995)
12. Arivazhagan, S., Ganesan, L., Subash Kumar, T.G.: A modified statistical approach for image fusion using wavelet transform. Springer, London (2008)
13. Li, S., Kwok, J.T.: Multi-focus image fusion using artificial neural networks. Pattern Recognition Letters 23, 985–997 (2002)
14. Sendur, L., Selesnick, I.W.: Bivariate Shrinkage Functions for Wavelet-Based Denoising Exploiting Interscale Dependency. IEEE Transactions on Signal Processing 50(11), 2744–2755 (2002)
15. Kuncheva, L.I.: Fuzzy Classifier Design. Springer, Heidelberg (2000)
16. Rockinger, O.: Image Sequence Fusion Using a Shift Invariant Wavelet Transform. In: ICIP, pp. 288–291 (1997)
17. Petrović, V., Xydeas, C.: Evaluation of image fusion performance with visible differences. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 380–391. Springer, Heidelberg (2004)
18. http://imagefusion.org

Unsupervised Object Discovery from Images by Mining Local Features Using Hashing

Gibran Fuentes Pineda, Hisashi Koga, and Toshinori Watanabe

Graduate School of Information Systems, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-si, Tokyo 182-8585, Japan

Abstract. In this paper, we propose a new methodology for efficiently discovering objects from images without supervision. The basic idea is to search for frequent patterns of closely located features in a set of images and consider a frequent pattern as a meaningful object class. We develop a system for discovering objects from segmented images. This system is implemented by hashing only. We present experimental results to demonstrate the robustness and applicability of our approach.

1 Introduction

Object recognition and discovery from images have been challenging problems in image analysis over the past decades. Typically, objects are represented by either a set of geometric elements such as cones, spheres, and planes (model-based), their contour (shape-based) or their appearance (appearance-based). Then, an object class is modeled by creating an approximate representation (generative models such as Bayesian network [1], and Non-Negative Factorization [2]) or defining an optimal decision boundary (discriminative models, e.g. boosting [3], and SVM [4]) from a set of given examples. In general, these methods scale poorly for very large databases because a) they require some kind of supervision, b) their performance is greatly affected by the high dimensionality of the object representation, and/or c) they are tailored to specific object classes (e.g. faces).

This work attempts to overcome these limitations by efficiently discovering objects from images without supervision. Our assumption is that an object consists of multiple components which are expressed as a set of local image features. To discover object classes without supervision we search for frequent patterns of closely located image features and consider one frequent pattern as a meaningful object class. By searching the known classes, the same approach can be further used for matching a query object with the most similar class, thus enabling the unification of modeling and matching. To simplify the complexity of our approach, we implement it completely by relying on a single technique, namely hashing. We show that hashing can be used to efficiently realize a variety of similarity judgments. Specifically, we demonstrate that the next three kinds of similarity judgments can be implemented by hashing: (1) standard distance-based similarity judgment, (2) distance-based similarity judgment considering the relative size, and (3) matching robust to small variations. By experiment,

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 978–985, 2009.

we prove that our approach can discover diverse object classes robustly against rotation and slide operations as well as small intraclass variations.

This paper is organized as follows. We describe locality-sensitive hashing in Sect. 2. In Sect. 3 we give an overview of our approach. Then, Sect. 4 discusses the detailed description of our system and reports some experimental results. Finally, Sect. 5 concludes the paper.

2 Similarity Judgments by Hashing

Similarity judgment is a fundamental element for pattern recognition in image analysis systems where multiple similarity measures might be necessary because items are defined by many attributes. However, as most image analysis schemes presume only specific spaces and similarity measures (commonly the Euclidean distance), it is not guaranteed that the same scheme will have the same good performance when applied to other spaces and/or similarity measures. The complexity of the system will increase if one decides to support several schemes simultaneously to treat different kinds of similarity measures.

Since hashing techniques have provided an efficient searching mechanism for various similarity judgments that are common in image analysis tasks, we believe that it is possible to construct a simple and efficient image analysis system by using such techniques. Hence, in this paper we consistently rely on hashing techniques inspired by the *locality-sensitive hashing* (LSH) [5].

We describe in detail LSH hereinafter. Let P be a set of points in a d-dimensional space and C be the maximum coordinate value of any point in P. We can transform every $p \in P$ to a Cd-dimensional vector by concatenating unary expressions for every coordinate, that is,

$$f(p) = \mathrm{Unary}(x_1)\mathrm{Unary}(x_2)\cdots\mathrm{Unary}(x_d), \tag{1}$$

where $\mathrm{Unary}(x)$ is a sequence of x ones followed by $C - x$ zeros. A hash function is computed by picking up k bits (which are called *sample bits*) randomly from these Cd bits and concatenating them. This corresponds to partitioning the d-dimensional space into cells of different sizes by k hyperplanes so that near points tend to have the same hash value. As k becomes large, remote points are less likely to take the same hash value because the size of generated cells becomes small. Figure 1 illustrates the space partitioning when $d = 2$, $C = 5$ and $k = 2$. This example presents the hash value of each cell when the second and eighth bits (i.e. x = 2 and y = 3) are selected from the total $2 \times 5 = 10$ bits. By contrast, depending on the result of space division, near points may take different hash values (e.g. point A and point B in Fig. 1). To exclude this failure, multiple l hash functions $h_1, h_2, \cdots h_l$ are prepared in LSH expecting that two points close to each other will take the same hash value at least for one hash function.

Overall, LSH considers that a pair of points with the same hash value are close to each other. We borrow this idea but while this LSH scheme utilizes randomized functions, we define deterministic functions more suitable for our object discovery scheme.

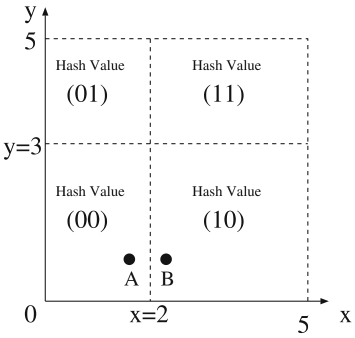


Fig. 1. Space partition by LSH

3 Overview of Our Approach

In this section we present an overview of our approach for discovering objects automatically from a set of images Σ. Each image in Σ consists of several local image features. The underlying idea is to search for frequent patterns of closely located features in Σ and consider each frequent pattern as a meaningful object class. Thereby, our approach runs in four phases described below.

Phase I: By extracting every feature in Σ, we derive a set of object components. We denote this set by $C = \{C_1, C_2, \ldots, C_N\}$.

Phase II: The components in C are classified according to their attributes. A label ID is assigned to each component according to the classification result; the labels are expressed by $\ell_1, \ell_2, \ldots, \ell_M$, where M is the number of component classes.

Phase III: Closely located components are gathered to generate object candidates. Let $T = \{T_1, T_2, \ldots, T_Z\}$ be the set of all object candidates.

Phase IV: We determine object classes by searching frequent patterns in T. A pattern with multiple occurrences is regarded as a meaningful object class. Each object class is represented by the set of component labels that is common in the multiple occurrences.

Figure 2 presents an example of the operation of our approach. First, 11 components from C_1 to C_{11} are extracted from two images. Next, the labels from ℓ_1 to ℓ_6 are assigned to each component; here, similar components have the same label (e.g. both roofs C_3 and C_8 have ℓ_3). Then, the object candidates T_1, T_2, T_3 and T_4 are generated by gathering closely located components. Finally, the Class 1 (”tree”) and Class 2 (”house”) are regarded as meaningful object classes because each of them has two occurrences in the images; ”tree” is represented by ℓ_1 and ℓ_2 whereas ”house” is represented by ℓ_3, ℓ_4 and ℓ_5.

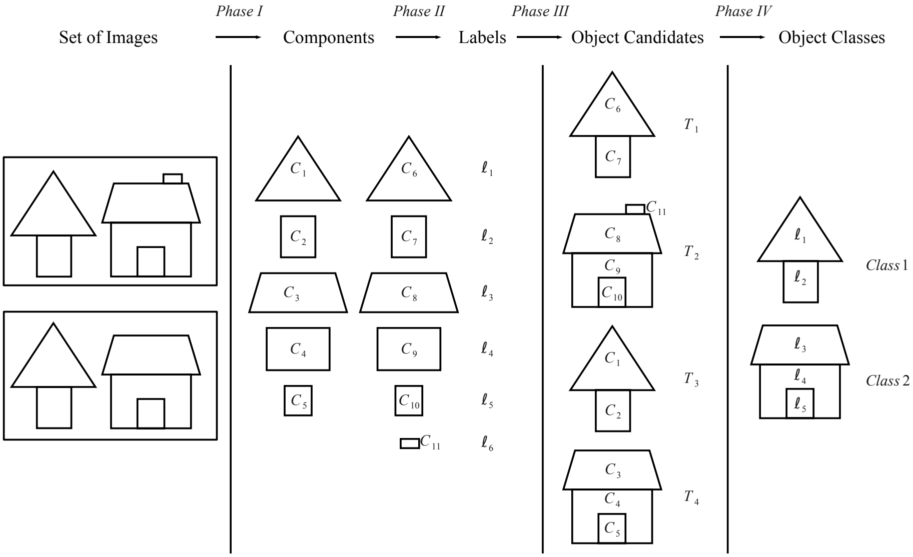


Fig. 2. Intuitive example of our object discovery approach

4 Object Discovery from Segmented Images

This section describes the implementation details of our system. As inputs, our system receives images preprocessed by segmentation and color quantization algorithms; each region of the preprocessed images is regarded as a single local feature. In addition, we assume that an object consists of closely located regions and does not overlap with other objects. Thus, our system performs three kinds of similarity judgments: (1) distance-based similarity judgment considering the relative size in Phase II, (2) ordinary distance-based similarity judgment in Phase III, and (3) matching robust to small variations in Phase IV. We modify the LSH scheme described in Sect. 2 to implement the first two kinds of similarity judgments whereas for the third kind we extend the standard hashing for exact matching to perform matching robust to small variations.

4.1 Phase I: Extraction of Components

In order to extract object components, we first identify regions in Σ that correspond to background. Then, we consider as object components all regions in Σ that are not identified as background . Let us denote this set of components by C. Although the discrimination between background and foreground regions is difficult and sometimes requires supervision, when the background is non-textured or represents the largest regions of the image, the automatic identification of the background becomes possible.

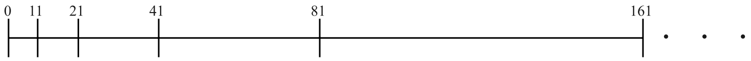


Fig. 3. Location of sample bits on the real line when $\alpha = 10$, $\beta = 2$ and $i = 1$

4.2 Phase II: Labeling of Components

The components in C are labeled according to their color and size such that components of the same color with similar size are assigned the same label. For this purpose, since color quantization is performed in the preprocessing phase, we may cluster components of different color separately. Therefore, the components of the same color are hashed according to their sizes. However, the similarity between sizes should be relative to their absolute value. Hence, the hash functions are defined by selecting sample bits at intervals proportional to the distance from the origin. That is, for the i-th hash function ($1 \leq i \leq l$), the sample bits are set as follows.

$$\begin{array}{c}\text{Location of sample bits}\\ h_i : \alpha + i,\, \alpha\beta + i,\, \alpha\beta^2 + i,\, \ldots\, , \alpha\beta^k + i,\end{array} \tag{2}$$

where α determines the position of the first sample bit and β is the growth factor of the intervals ($\alpha > 0$ and $\beta > 1$). Figure 3 illustrates the location of the sample bits on the real line when $\alpha = 10$, $\beta = 2$ and $i = 1$. Note that the intervals between the sample bits become wider as they become farther from 0.

To cluster similar components we apply the CENTER algorithm [6]. CENTER makes graphs where vertices are components and an edge is made between a pair of components if they have the same hash value at least for one hash function. Then, graphs are partitioned in such a way that in each cluster the center node has an edge to the other nodes. This process is carried out by following the next steps.

Step I: For each color, pick up the biggest unchecked component B in C.
Step II: Select all the unchecked components that have an edge to B and merge them into the same cluster.
Step III: Mark all the merged components as checked.
Step IV: Repeat step 1-3 until all the components have been checked.

After this process, we assign the labels $\ell_1, \ell_2, \ldots, \ell_M$ to the clusters according to the size of the center components such that ℓ_1 and ℓ_M corresponds to the largest and smallest component respectively.

4.3 Phase III: Generation of Object Candidates

We generate object candidates by clustering closely located components. The nearness between two separate components is determined by the Euclidean distance among their pixels. Therefore, we hash all pixels in every component in C. The hash functions are defined by selecting sample bits at equal intervals of a parameter I. Consequently, the number of sample bits k is expressed as follows.

$$k = \frac{X_{max} + Y_{max}}{I}, \tag{3}$$

where X_{max} and Y_{max} denote respectively the number of columns and rows of the given image. We define the I hash functions so that the sample bits do not coincide one another at all in the following way.

$$\begin{array}{l} \quad\quad\text{Location of sample bits} \\ h_1 : 1,\, I+1,\, 2I+1,\, \cdots\, (k-1)I+1 \\ h_2 : 2,\, I+2,\, 2I+2,\, \cdots\, (k-1)I+2 \\ \quad\;\vdots \\ h_I : I,\quad 2I,\quad\; 3I,\quad \cdots \quad\;\; kI \end{array}. \tag{4}$$

For generating object candidates, we adopt the next rule.

Rule 1. *Two separate components C_i and C_j $(i, j = 1, \ldots, N)$ are clustered into the same object candidate if one pixel in C_i and one pixel in C_j have the same hash value at least for one hash function.*

Each object candidate T_i $(1 \leq i \leq Z)$ is represented by a vector

$$T_i = [\upsilon_1, \ldots, \upsilon_M], \tag{5}$$

where υ_r $(1 \leq r \leq M)$ denotes the number of components with label ℓ_r in the object candidate T_i. For example, the object candidate T_1 in Fig. 2 is generated from the components C_6 and C_7 (with labels ℓ_1 and ℓ_2 respectively) because they are close to one another. In this case, the representation of the object candidate becomes $T_1 = (1, 1, 0, 0, 0, 0)$.

4.4 Phase IV: Determination of Object Classes

In order to determine meaningful object classes, we search for multiple occurrences of similar object candidates. We judge object candidates as similar if their primary components are the same. Standard hashing is applied to accelerate this process. To compute the hash value for an object candidate T_i $(1 \leq i \leq Z)$, we first concatenate the elements υ_r $(1 \leq r \leq M)$ of T_i, that is,

$$cat(T_i) = \upsilon_1 \upsilon_2 \cdots \upsilon_M, \tag{6}$$

where $\upsilon_1, \upsilon_2, \cdots, \upsilon_M$ are expressed by λ bits so that $|cat(T_i)| = \lambda M$. In order to avoid small intra-class variations, we generate J hash values for T_i by ignoring the $\xi, \xi+1, \ldots, \xi+J-1$ smallest components from $cat(T_i)$, where ξ presents the maximum integer such that the sum of the size of the ξ smallest components in T_i does not exceed the μ% of the whole size of T_i. After computing the J hash values for each object candidate, we obey the next rule to determine meaningful object classes.

Rule 2. *Two object candidates are classified into the same cluster if at least one of their J hash values is the same.*

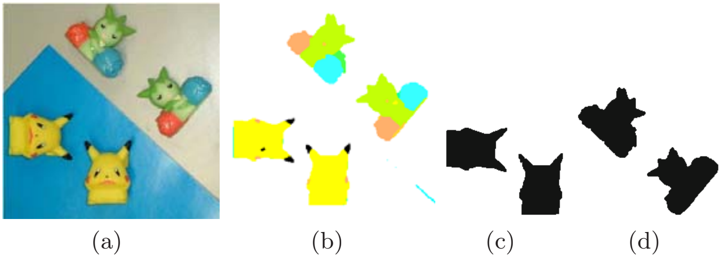


Fig. 4. Discovery of objects with rotation and slide variations: (a) original image, (b) preprocessed image, (c) class 1 and (b) class 2

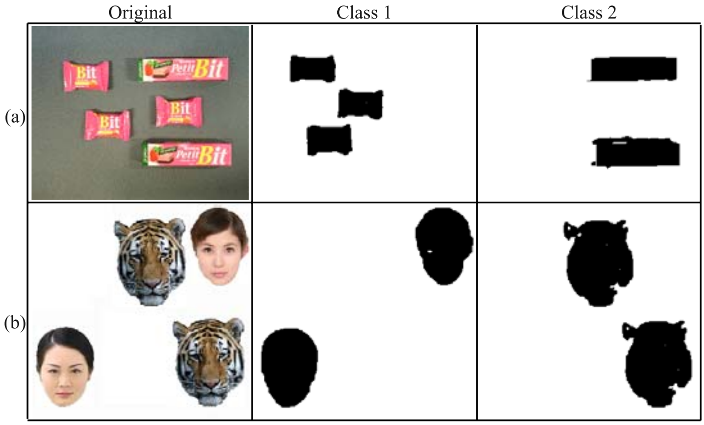


Fig. 5. Discovery of objects with intraclass variations

After clustering the object candidates, we regard as meaningful object classes only those clusters with multiple object candidates. We represent each of these classes in the same form as (5), where υ_r $(r = 1, \ldots, M)$ stands for the number of components with label ℓ_r that are common to all object candidates of the given class. For instance, T_2 and T_4 in Fig. 2 are classified into the same cluster by ignoring C_{11}, which is extremely small relative to the size of T_2. Then, since this cluster has two object candidates, it is regarded as a meaningful object class (Class 1) and represented by ℓ_3, ℓ_4 and ℓ_5, i.e., $\upsilon = (0, 0, 1, 1, 1, 0)$. Note that ℓ_6 is not included, because it is not a component of T_4.

4.5 Experimental Results

For the experiments, each image was segmented by using the MST-based algorithm [7] and then a color quantization was performed. Finally, we considered the extremely big regions of the image as background and remove them so that objects were isolated. An example of the segmentation, quantization and background removal can be seen in Fig. 4(a) and 4(b).

Initially, we evaluated the robustness of our system against rotation and slide operations. To that end, we applied our system to an image that contains two instances of two different object classes (Fig. 4(b)). Note that the orientations of the two instances of the same class differ approximately by 90 degrees. Since our system does not consider the exact location relation between components, both object classes (Fig. 4(c) and 4(d)) were successfully discovered despite these transformations.

We also evaluated the robustness of our system against intraclass variations. In Fig. 5 we present two examples of this evaluation. The first example consists of an image containing two kinds of candy (Fig. 5(a)). The other example consists of two kinds of faces: human faces and tiger faces (see Fig. 5(b)). Note that the two human faces are different. In both examples our system derived two object classes successfully. The columns Class 1 and Class 2 in Fig. 5 illustrate the instances of each derived class in each image. As we can observe our system can cope with small intraclass variations such as faces of different subjects.

5 Conclusions

We proposed a new methodology for discovering object classes from images which can discover and recognize diverse object classes without supervision. This methodology can be suitable for indexing and searching objects in large image databases with diverse contents. We demonstrated that frequent patterns of local image features can lead to discover meaningful object classes. Our approach can be completely implemented by only hashing which simplifies its implementation and at the same time enables the integration of various similarity measures. We proved by experiment that our approach is robust against rotation and slide operations as well as small intraclass variations.

References

1. Bar-Hillel, A., Weinshall, D.: Efficient learning of relational object class models. International Journal of Computer Vision 77(1–3), 175–198 (2008)
2. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
3. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(3), 416–431 (2006)
4. Heisele, B., Serre, T., Poggio, T.: A component-based framework for face detection and identification. International Journal of Computer Vision 74(2), 167–181 (2007)
5. Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: Thirtieth Annual ACM Symposium on the Theory of Computing, pp. 604–613 (1998)
6. Haveliwala, T.H., Gionis, A., Indyk, P.: Scalable techniques for clustering the web. In: Third International Workshop on the Web and Databases, pp. 129–134 (2000)
7. Tsunoda, N., Watanabe, T., Sugawara, K.: Image segmentation by adaptive thresholding of minimum spanning trees. Transactions of the Institute of Electronics, Information and Communication Engineers J87-D-II(2), 586–594 (2004)

XVIII CASI 2009 Workshop I: Intelligent Computing for Remote Sensing Imagery

Spectral Estimation of Digital Signals by the Orthogonal Kravchenko Wavelets $\left\{\widetilde{h_a(t)}\right\}$

Victor Kravchenko[1], Hector Perez Meana[2], Volodymyr Ponomaryov[2],
and Dmitry Churikov[1]

[1] Kotel'nikov Institute of Radio Engineering and Electronics of RAS,
Mokhovaya 11, build. 7, 125009, Moscow, Russia
olegk@lianet.ru, mpio_nice@mail.ru
[2] National Polytechnic Institute of Mexico, AV. SANTA ANA No.1000,
Col. San Francisco Culhuacan 04430, Mexico-city, Mexico
hmperezm@ipn.mx, vponomar@ipn.mx

Abstract. In this article, the approach based on the orthogonal Kravchenko wavelets $\left\{\widetilde{h_a(t)}\right\}$ is proposed. There is shown that obtained structures have some advantages in comparison with spectral wave analysis of ultra wideband (UWB) signals that are widely used in the remote sensing. This approach based on application of wavelets as spectral kernels is considered in the problems of digital UWB signal processing. In communication theory, the signals are represented in the form of linear combination of elementary functions. Application of spectral analysis of UWB signals in basis of digital functions in comparison with the spectral harmonious analysis gives certain advantages which are defined by the physical nature of the signal representation. Optimal processing in spectral area in comparison with the time possesses has advantages on application of numerical algorithms. The physical characteristics and analysis of numerical experiment confirm efficiency of new wavelets in the spectral estimation and digital UWB signal processing.

Keywords: Atomic functions, Wavelets, Remote sensing, Digital ultra wideband signal processing.

1 Introduction

Application of the spectral analysis of signals in basis of digital functions in comparison with the spectral harmonious offers certain advantages which are defined by the physical nature of representation of the signals [1-11] in the remote sensing problems [1, 2]. Signals can be set in the form of some linear combination of elementary functions [3-5]

$$s(t)=\sum_{k=0}^{N-1}C(k)\varphi(k,t), \tag{1}$$

where $\varphi(k,t)$ is an elementary function of number k, and N is quantity of the functions used in the decomposition. At approximation the generalized Fourier transformation of a kind

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 989–996, 2009.

$$C(k)=\int_0^T s(t)\varphi(k,t)dt \tag{2}$$

provides the minimum value of mean-square error. Thus, there is signal decomposition on some basis, which in many problems of digital signal processing can not be orthogonal. The work purpose is consisted of justify the advantages of the new Kravchenko wavelets.

2 Atomic Functions $h_a(t)$ and Their Properties

Atomic functions [5-8], [12-17] $h_a(t)$ $(a > 1)$ are finite decisions of the functional-differential equation

$$y'(t)=\frac{a^2}{2}\left(y(at+1)-y(at-1)\right), \tag{3}$$

where a is any real number. The basic properties of $h_a(t)$ are the following.

1^{O}. Compact support: $\left[-\frac{1}{a-1};\frac{1}{a-1}\right]$. 2^{O}. $h_a(t)=\frac{a}{2}$ for $t\in\left[-\frac{a-2}{a(a-1)};\frac{a-2}{a(a-1)}\right]$, $a\geq 2$. 3^{O}. Position of inflection $t=\left[\mp\frac{1}{a},\frac{a}{4}\right]$. 4^{O}. The Fourier transform of $h_a(t)$ looks like

$$\hat{h}_a(\omega)=\prod_{k=1}^{\infty}\text{sinc}\left(\frac{\omega}{a^k}\right). \tag{4}$$

In practical calculations, it is enough to limit the product (4) to a small number of terms, as they quickly aspire to unit with growth of k.

5^{O}. Derivatives of $h_a(t)$ are expressed through shifts-compression of the function recurrently by means of a parity (3).

3 Constructing of the Orthogonal Kravchenko Wavelets

The orthogonal Kravchenko wavelets that is based on AF $h_a(t)$ and have smooth Fourier transform are proposed. It allows providing the best time localization in comparison with Kotelnikov-Shannon wavelet. Their construction [7-11] is carried out by means of the quadrature mirror filters $\widehat{m_0}(\omega)$. For maintenance of orthogonality performance, the following conditions in transitive area are necessary

$$\left|\widehat{m_0}(\omega)\right|^2+\left|\widehat{m_0}(\omega-2\pi)\right|^2=2\,. \tag{5}$$

The Fourier transform of scaling function [7-11] is defined from the equation

$$\hat{\varphi}(\omega)=\frac{1}{\sqrt{2}}\hat{m}_0\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right) \Leftrightarrow \hat{\varphi}(\omega)=\prod_{k=1}^{\infty}\frac{1}{\sqrt{2}}\hat{m}_0\left(\frac{\omega}{2^k}\right). \tag{6}$$

The Fourier transform of wavelet function can be written as

$$\hat{\psi}(\omega)=\frac{1}{\sqrt{2}}\hat{g}\left(\frac{\omega}{2}\right)\hat{\varphi}\left(\frac{\omega}{2}\right), \tag{7}$$

where $\hat{g}(\omega)=\mathrm{e}^{-i\omega}\hat{m}_0(\omega+\pi)$. Considering the properties of $h_a(t)$ it is possible to simplify a construction of function $\hat{m}_0(\omega)$. For this purpose, we modify $h_a(t)$ according to following additional conditions.

6$^{\mathrm{O}}$. The support of function $t\in\left[-\frac{2}{3}\pi;\frac{2}{3}\pi\right]$. 7$^{\mathrm{O}}$. $\tilde{h}_a(t)=\sqrt{2}$ for $t\in\left[-\frac{1}{3}\pi;\frac{1}{3}\pi\right]$.

8$^{\mathrm{O}}$. For interface of the Fourier transforms $\tilde{h}_a\left(\frac{1}{2}\pi\right)=1$.

Thus, as $\hat{m}_0(\omega)$ we take $\tilde{h}_a(t)$ with formal replacement of argument $t=\omega$. Properties 1st and 7th allow rewriting (7) in such a form

$$\hat{\varphi}(\omega)=\tilde{h}_a\left(\frac{\omega}{2}\right), \tag{8}$$

so long as $\tilde{h}_a\left(\frac{\omega}{2}\right)\neq 0$ only for $t\in\left[-\frac{4}{3}\pi;\frac{4}{3}\pi\right]$, where $\tilde{h}_a\left(\frac{\omega}{4}\right)=\sqrt{2}$. From 5th property, it follows that in transitive area $h_a(t)$ is symmetric relatively of inflection. Therefore, if as function $\tilde{h}_a(\omega)$ we take a square root of the $h_a(t)$, then the condition (4) is carried out. Hence, the function $\tilde{h}_a(\omega)$ is constructed as follows:

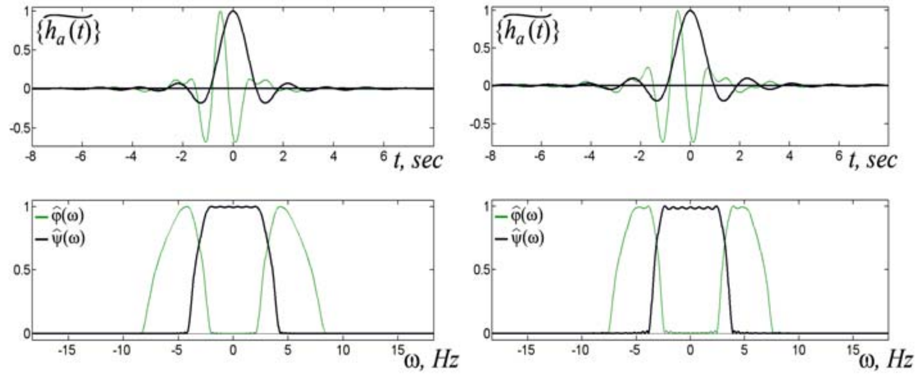
(a) replacement of variable $t=\frac{3}{2\pi}\omega$, (b) to satisfy the property 6th it should be $a\geq 4$, (c) scaling of function and argument, and also taking a square root.

Thus, we obtain the function for the scaling equation with various velocities of increase and recession depending of parameter a

$$\tilde{h}_a(\omega)=\frac{2}{\sqrt{a}}\sqrt{h_a\left(\frac{2}{a\pi}\omega\right)}. \tag{9}$$

Finally, we can write the Fourier transforms for scaling and wavelet functions as follows:

$$\hat{\varphi}(\omega)=\frac{2}{\sqrt{a}}\sqrt{h_a\left(\frac{\omega}{a\pi}\right)},\ \hat{\psi}(\omega)=\mathrm{e}^{-i\omega}\frac{2\sqrt{2}}{a}\sqrt{h_a\left(\frac{2}{a\pi}(\omega+\pi)\right)h_a\left(\frac{\omega}{a\pi}\right)}. \tag{10}$$



a) *b*)

Fig. 1. The Kravchenko $\left\{\widetilde{h_a(t)}\right\}$ scaling (dark line) and wavelet (light line) functions for $a = 4$ (*a*) and $a = 6$ (*b*)

As an example, the Kravchenko wavelets $\left\{\widetilde{h_a(t)}\right\}$ and their spectrums are exposed in Fig.1 for $a = 4, 6$.

Energy of scaling and wavelet functions on N^{th} level is equal to energy of scaling function of $(N+1)^{\text{th}}$ level because

$$\left|\hat{\varphi}(\omega)\right|^2 + \left|\hat{\psi}(\omega)\right|^2 = \left|\hat{m}_0\left(\frac{\omega}{4}\right)\right|^2 = \left|\hat{\varphi}\left(\frac{\omega}{2}\right)\right|^2. \tag{11}$$

Constructed functions are satisfying with all wavelet properties [6-10] and also they are orthogonal ones.

4 Physical Characteristics

To study the scaling and wavelet functions, we use the modified physical characteristics [5]: *wideband index* μ; *central frequency of spectral density function* (SDF) f_0; *relative position of SDF maximum* (defined as $\gamma_1 = f_m / f_0$, where f_m is frequency of the main maximum); *relative position of SDF first zero* γ_2; *relative SDF width on level -3 dB* γ_3; *relative SDF width on level -6 dB* γ_4; *coherent amplification* defined as $\gamma_7 = \frac{1}{\tau}\int\limits_{-\tau/2}^{\tau/2} \left|\varphi(t)\right| dt$; *equivalent noise band* γ_8; *maximum level of sidelobes* (in dB) γ_9. Mentioned physical characteristics of the Kravchenko wavelets and known ones for comparison are exposed in Table 1 below.

Table 1. Physical characteristics of the Kravchenko wavelets in comparison with known ones

	a	γ_1	γ_2	γ_3	γ_4	γ_7	γ_8	γ_9
Kravchenko wavelets $\left\{\widetilde{h_a(t)}\right\}$								
$\varphi(t)$	4	0	0,676	0,563	0,608	0,704	4,158	-19,65
$\psi(t)$		0,850	0,665	0,688	0,836	0,810	4,032	-19,77
$\varphi(t)$	5	0	0,637	0,546	0,585	0,655	3,744	-17,36
$\psi(t)$		0,874	0,577	0,642	0,762	0,713	3,442	-17,30
$\varphi(t)$	6	0	0,619	0,534	0,568	0,629	3,505	-16,73
$\psi(t)$		0,876	0,528	0,614	0,710	0,671	3,230	-16,55
Kotelnikov-Shannon wavelet								
$\varphi(t)$	-	0	0,540	0,494	0,512	0.500	2,429	-11,73
$\psi(t)$	-	0,996	0,290	0,489	0,523	0,458	2,037	-10,56
Meyer wavelet								
$\varphi(t)$	-	0	0,654	0,523	0,557	0,586	3,430	-29,11
$\psi(t)$	-	0,863	0,648	0,574	0,671	0,565	3,196	-27,13
Daubechies 4 wavelet								
$\varphi(t)$	-	0	0,500	0,284	0,329	0,352	2,485	-10,92
$\psi(t)$	-	0,847	0,637	0,375	0,506	0,343	2,362	-10,63

5 Models of Ultra Wideband Signals

We consider of following models of ultra wideband (UWB) signals [3, 4], [5-7] in respect of remote sensing problems [1, 2]:

1. $y_1(t) = -\left(H\left(\frac{t+0,5}{\tau}\right) - H\left(\frac{t-0,5}{\tau}\right)\right)\cdot \operatorname{sgn}(t)$,

2. $y_2(t) = (-1)^n \cdot \sin\left(\pi n \frac{t}{\tau}\right)\cdot \exp\left(-\left|\frac{t}{\tau}\right|\right)\cdot\left(H\left(\frac{t}{\tau}+1\right) - H\left(\frac{t}{\tau}-1\right)\right)$,

3. $y_3(t) = -\frac{2t}{\tau^2}\cdot\exp\left(-\left(\frac{t}{\tau}\right)^2\right)$, 4. $y_4(t) = -\frac{2}{\tau^2}\left(1-\frac{2t^2}{\tau^2}\right)\cdot\exp\left(-\left(\frac{t}{\tau}\right)^2\right)$,

5. $y_5(t) = \exp\left(-\left(\frac{t}{2\tau}\right)^2\right) n! \sum_{k=0}^{[n/2]}\left(-\frac{1}{2}\right)^k \frac{(t/\tau)^{n-2k}}{(n-2k)!k!}$,

where $H(t)=\begin{cases}0, & t<0,\\ 1, & t\geq 0.\end{cases}$ is the Heaviside function and $\operatorname{sgn}(t)$ is the sign function. Their physical characteristics are presented in Table 2.

Table 2. Model UWB signals and their physical characteristics

No.	Model UWB signals	μ	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8	γ_9
1	$y_1(t)$, τ=1 s	1,54	0,91	1,04	0,85	1,11	-1,58	-1,41	1,00	1,00	-5,29
2	$y_1(t)$, τ=1,5 s	1,50	0,92	1,00	0,83	1,11	-1,49	-1,34	0,67	0,67	-5,28
3	$y_2(t)$, n=1	1,46	0,91	1,06	0,81	1,13	-11,47	-7,81	0,81	0,63	-12,99
4	$y_3(t)$, τ=0,3 s	1,59	0,87	2,26	0,81	1,13	-37,58	-9,17	1,43	1,04	-49,15
5	$y_3(t)$, τ=0,5 s	1,48	0,87	1,82	0,88	1,18	-8,80	-5,88	0,87	6,49	-18,87
6	$y_4(t)$, τ=0,3 s	1,22	0,90	2,19	0,84	1,16	-32,05	-8,43	1,94	1,07	-43,44
7	$y_4(t)$, τ=0,5 s	1,20	1,00	1,11	0,83	1,17	-6,20	-4,60	1,22	0,69	-14,08
8	$y_5(t)$, n=1, τ=0,15 s	1,59	0,87	2,26	0,81	1,13	-37,58	-9,17	1,43	1,04	-49,15
9	$y_5(t)$, n=3, τ=0,15 s	1,77	0,85	1,39	0,89	1,09	-25,77	-10,44	1,15	0,81	-34,63
10	$y_5(t)$, n=5, τ=0,15 s	1,81	0,84	1,08	0,93	1,07	-17,01	-10,40	1,02	0,70	-26,63
11	$y_5(t)$, n=1, τ=0,1 s	1,59	0,86	3,28	0,85	1,15	-96,77	-8,86	2,14	1,57	-108,3
12	$y_5(t)$, n=3, τ=0,1 s	1,75	0,82	2,21	0,90	1,10	-81,50	-10,37	1,73	1,22	-92,79
13	$y_5(t)$, n=5, τ=0,1 s	1,83	0,85	1,69	0,93	1,06	-68,61	-11,34	1,53	1,04	-77,97
14	$y_5(t)$, n=7, τ=0,1 s	1,85	0,87	1,42	0,95	1,06	-57,50	-11,65	1,40	0,93	-67,70

6 Quality Functional of Wavelet-Basis Choice for Analysis of UWB Signals

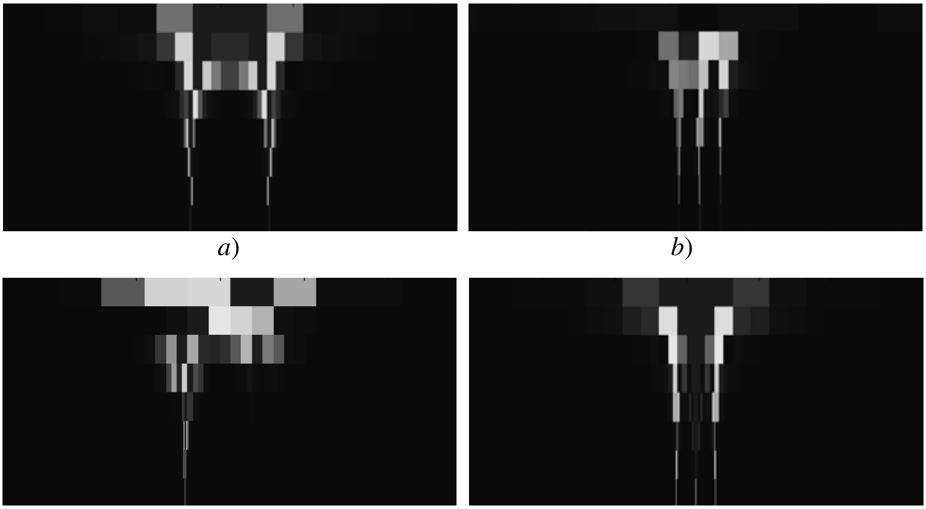
It is proposed to apply the quality functional in the analysis of the UWB signals allowing optimal choice of basic wavelets functions in such a form

$$J(\psi, y)=\sum_{k=0}^{N}\left|\frac{\gamma_k^{\psi}-\gamma_k^{y}}{\gamma_k^{y}}\right|^2, \tag{12}$$

where $\psi(t)$ is wavelet function, $y(t)$ is analyzed signal, γ_k^{ψ} and γ_k^{y} are their physical characteristics, and N is quantity of compared parameters. Here, $\gamma_0=\mu$ and $N=4$. Below, in Table 3, the values of quality functional of model UWB signal processing for the Kravchenko $\left\{\widetilde{h_2(t)}\right\}$ and Meyer wavelets are presented.

Table 3. The values of quality functional of model UWB signal processing for the Kravchenko $\left\{\widetilde{h_2(t)}\right\}$ and Meyer wavelets

No. of UWB signals realization	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Kravchenko wavelet	**29,63**	**29,92**	**29,77**	**68,43**	**49,55**	**43,77**	**12,02**	**68,43**	**106,7**	**132,1**	**88,58**	**131,1**	**143,9**	**158,9**
Meyer wavelet	35,15	35,63	35,57	71,48	52,28	45,20	15,60	71,48	113,6	142,3	92,44	137,7	151,6	167,9



a)

b)

c)

d)

Fig. 2. Discrete wavelet coefficients of 1st (*a*), 3th (*b*), 4th (*c*), and 7th (*d*) UWB signals realization for the Kravchenko $\left\{\widetilde{h_2(t)}\right\}$ wavelet

7 Conclusions

In this article, the application of the orthogonal Kravchenko wavelets $\left\{\widetilde{h_a(t)}\right\}$ for the digital UWB signal processing is proposed and justified. A number of the test signals and their wavelet transform examples are presented. A numerical experiment carried out and also an analysis of physical results have shown the advantages of novel wavelet class in its applications for areas of remote sensing, radar, computer vision, radio physics, etc.

Acknowledgments. The investigations were supported by grant NSh-5708.2008.9 “The new methods in some problems of acoustooptics, radio physics and medicine on the basis of atomic functions, wavelets and fractals”.

References

1. Soumekh, M.: Synthetic Aperture Radar Signal Processing with MATLAB Algorithms. John Wiley & Sons, Inc., NY (1999)
2. Cumming, I.G., Wong, F.H.: Digital Processing of Synthetic Aperture Radar Data: Algorithms and Implementation. Artech house, Inc. Norwood (2005)
3. Ghavami, M., Michael, L.B., Kohno, R.: Ultra Wideband Signals and Systems in Communication Engineering. John Wiley & Sons, Ltd., Chichester (2004)
4. Arslan, H., Chen, Z.N., Benedetto, M.G.: Ultra Wideband Wireless Communication. John Wiley & Sons, Ltd., Chichester (2006)
5. Kravchenko, V.F.: Lectures on the Theory of Atomic Functions and Their Some Applications. Publishing House Radio Engineering, Moscow (2003)
6. Kravchenko, V.F., Rvachov, V.L.: Boolean Algebra, Atomic Functions and Wavelets in Physical Applications. Fizmatlit, Moscow (2006)
7. Kravchenko, V.F. (ed.): Digital Signal and Image Processing in Radio Physical Applications. Fizmatlit, Moscow (2007)
8. Kravchenko, V.F., Labun’ko, O.S., Lerer, A.M., Sinyavsky, G.P.: Computing Methods in the Modern Radio Physics. In: Kravchenko, V.F. (ed.) Fizmatlit, Moscow (2009)
9. Meyer, Y.: Wavelets and Operators. Cambridge University Press, Cambridge (1992)
10. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial & Applied Mathematics, U.S. (1992)
11. Mallat, S.G.: A Wavelet Tour of Signal Processing. Academic Press, NY (1998)
12. Kravchenko, V.F., Churikov, D.V.: Digital Signal and Image Processing on Basis of Orthogonal Kravchenko Wavelets. In: Proc. Int. Conference “DAYS on DIFFRACTION”, May 26-29, pp. 53–54. St.Petersburg, Russia (2009)
13. Kravchenko, V.F., Churikov, D.V.: Atomic Functions $h_a(x)$ and New Orthogonal Wavelets on Their Basis. Successes of Modern Radio Electronics 6, 67–88 (2008)
14. Kravchenko, V.F., Churikov, D.V.: A New Class of Orthogonal Kravchenko WA-system Functions $\left\{\widetilde{h_a(t)}\right\}$. Telecommunications and Radio Engineering 68(8), 649–666 (2009)
15. Gulyaev, Y.V., Kravchenko, V.F., Pustovoit, V.I.: A New Class WA-systems of Kravchenko-Rvachev Functions. Doklady Mathematics 75(2), 325–332 (2007)
16. Kravchenko, V.F., Churikov, D.V.: A New Class of Orthogonal Kravchenko Wavelets. In: Proceedings of International Conference RVK 2008 and MMWP 2008, Växjö, Sweden, June 9-13, pp. 39–43 (2008)
17. Gomeztagle, F., Kravchenko, V.F., Ponomaryov, V.I.: Super-resolution Method Based on Wavelet Atomic Functions in Images and Video Sequences. Telecommunications and Radio Engineering 68(9), 747–761 (2009)

Video Denoising by Fuzzy Directional Filter Using the DSP EVM DM642

Francisco J. Gallegos-Funes[1], Victor Kravchenko[2],
Volodymyr Ponomaryov[1], and Alberto Rosales-Silva[1]

[1] National Polytechnic Institute of Mexico
vponomar@ipn.mx, arosaless@ipn.mx, fgallegosf@ipn.mx
[2] Institute of Radio Engineering and Electronics, Moscow, Russia
olegk@lianet.ru

Abstract. We present a new 3D Fuzzy Directional (3D-FD) algorithm for the denoising of video colour sequences corrupted by impulsive noise. The proposed approach consists of the estimations of movement levels, noise in the neighborhood video frames, permitting to preserve the edges, fine details and chromaticity characteristics in video sequences. Experimental results show that the noise in these sequences can be efficiently removed by the proposed 3D-FD filter, and that the method outperforms other state of the art filters of comparable complexity on video sequences. Finally, hardware requirements are evaluated permitting real time implementation on DSP EVM DM642.

Keywords: Fuzzy logic, Directional Processing, Impulsive Noise.

1 Introduction

Video signals are corrupted by noise from the capturing devices or during transmission due to random thermal or other electronic noises. Noise reduction can considerably improve visual quality and facilitate the subsequent processing tasks, such as video compression. There are many existing video denoising approaches employed for impulsive noise suppression in images and video sequences [1-11]. Another problem, which is exist here, is possible camera moving or occlusion that produce the temporal changes, which together spatial fine details and edges, and texture hinder traditional technique and demand to introduce novel adaptive frameworks.

The proposed 3D-FD filter is based on fuzzy set theory and order vector statistic technique in processing of RGB colour video sequences, detecting the noise and movement levels, and permitting to suppress impulsive noise. The 3D-FD algorithm consists of use of novel 2D-FD filter as a first (spatial) stage applied to *t*-1 frame. At the second (temporal) stage of algorithm, the filtering result of first stage should be employed in filtering of next *t* frame of the video sequence. As a final stage, the present frame is filtered applying again the 2D-FD filter, which permits noise suppression in a current frame. Numerical simulations demonstrate that new 3D-FD filter can outperform several filtering approaches in processing the video colour sequences in terms of noise suppression, edge and fine detail preservation, and colour retention.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 997–1004, 2009.

Finally, the Real-Time evaluation was realized using Texas Instruments EVM DM642 presenting good capabilities in the real-time environment.

2 Proposed 3D Fuzzy Directional (3D-FD) Filter

The 3D Fuzzy Directional (3D-FD) procedure employs a 2D-FD filter as a first (spatial) stage in the initial frame of video sequence. After, the temporal stage of the algorithm, proposed 2D-FD algorithm should be used again to suppress the non-stationary noise left during the temporal stage of the procedure.

Let introduce *gradients* and *angle variance* as absolute differences to represent the level of similarity among different pixels. Next, we calculate the gradient for each direction $\gamma=\{N,E,S,W,NW,NE,SE,SW\}$ according **to** Figure 1a. We employ not only one *basic gradient* for any direction, but also four *related gradient*, with (k,l) values $\{-2,-1,0,1,2\}$ [12,13]. The *angle variance* is computed for each a channel in such a way where we omit two of the three channels in the case of each a RGB colour frame. We use two neighbor frames of a video sequence to calculate the movement and noise fuzzy levels of a central pixel. A 5x5x2 sliding window is formed by *past* and *present* frames (see Figure 1b),

The *gradient* can be computed as,

$$G^{\beta}_{(k,l)} = \left| A^{\beta}_{(i+k,j+l)} - B^{\beta}_{(i+k,j+l)} \right|, \tag{1}$$

where $G^{\beta}_{(k,l)}$ can be a gradient ∇^{β}_{γ} or a *gradient difference value* $\lambda^{\beta}_{(k,l)}$ (in the case of 3D), $A^{\beta}_{(i+k,j+l)}$ and $B^{\beta}_{(i+k,j+l)}$ are the pixels in 2D window processing (see Figure 1a) or the pixels in *t*-1 and *t* frames of sequence (see Figure 1b), β is the RGB colour space, and γ marks any chosen direction according to indexes $(k,l)\in\{-2,-1,0,1,2\}$.

The *angle variance* is calculated as follows:

$$\varphi^{\beta}_{(k,l)} = \arccos\left[\frac{2(255)^2 + A^{\beta}_{(i+k.j+l)} \cdot B^{\beta}_{(i+k,j+l)}}{\left(2(255^2) + \left(A^{\beta}_{(i+k,j+l)}\right)^2\right)^{1/2} \cdot \left(2(255^2) + \left(A^{\beta}_{(i+k,j+l)}\right)^2\right)^{1/2}} \right]. \tag{2}$$

where $\varphi^{\beta}_{(k,l)}$ can be an angle variance $\theta^{\beta}_{(k,l)}$ or an absolute difference vectorial (angle variance in the case of 3D) value $\phi^{\beta}_{(k,l)}$.

Figure 1a exposes the employed pixels in 2D processing procedure in the case of *SE* direction for the *basic* and *related* components. The basic gradient value for *SE* direction is $\nabla^{\beta}_{(1,1)}x(i,j)=\nabla^{\beta}_{SE(b)}$ and the related gradients and angle variance values are given by $F^{\beta}_{(0,2)}x(i-1,j+1)=\nabla^{\beta}_{SE(r_1)}=\theta^{\beta}_{SE(r_1)}$, $F^{\beta}_{(2,0)}x(i+1,j-1)=\nabla^{\beta}_{SE(r_2)}=\theta^{\beta}_{SE(r_2)}$, $F^{\beta}_{(0,0)}x(i-1,j+1)=\nabla^{\beta}_{SE(r_3)}=\theta^{\beta}_{SE(r_3)}$, and $F^{\beta}_{(0,0)}x(i+1,j-1)=\nabla^{\beta}_{SE(r_4)}=\theta^{\beta}_{SE(r_4)}$.

In the case of 3D procedure, we calculate the *absolute difference gradient values* $\nabla^{t\beta}_{\gamma}$ of a central pixel with respect to its neighbours for a 5x5x1 window processing.

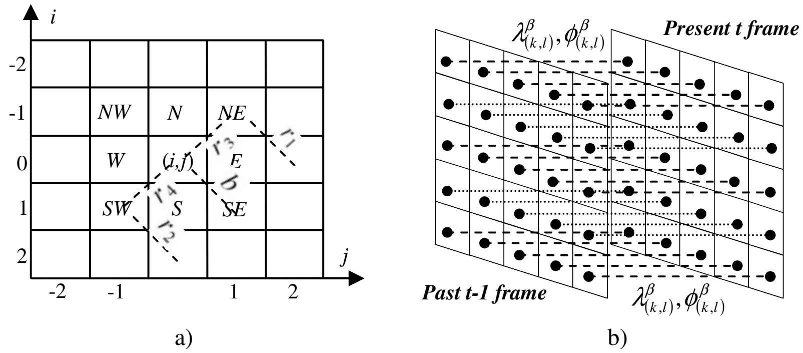

i
-2
-1
0
1
2
NW
N
NE
W
(i,j)
E
SW
S
SE
r1
r2
r3
r4
b
j
-2
-1
1
2
$\lambda^{\beta}_{(k,l)}, \phi^{\beta}_{(k,l)}$
Present t frame
Past t-1 frame
$\lambda^{\beta}_{(k,l)}, \phi^{\beta}_{(k,l)}$
a)
b)

Fig. 1. Windows processing, a) Neighbourhood pixels, *basic* (*b*) and *related* (r_1, r_2, r_3, r_4) directions for *gradient* and *angle variance* values, and b) Processing frames in the proposed 3D-FD filter

Using *angle variance value* $\phi^{\beta}_{(k,l)}$, we can characterize the *absolute vectorial variance* $\nabla^{tt\beta}_{\gamma}$. The *absolute difference gradient value* and *absolute vectorial variance* for the *SE* (*basic*) direction are given by values $\nabla^{t\beta}_{SE(basic)} = \nabla^{t\beta}_{(1,1)}\lambda_{(0,0)} = \left|\lambda^{\beta}_{(0,0)} - \lambda^{\beta}_{(1,1)}\right|$ and $\nabla^{tt\beta}_{SE(basic)} = \nabla^{tt\beta}_{(1,1)}\phi_{(0,0)} = \left|\phi^{\beta}_{(0,0)} - \phi^{\beta}_{(1,1)}\right|$, respectively. The same reasoning done by $\nabla'^{\beta}_{SE(basic)}$ with respect to $\nabla^{\beta}_{SE(basic)}$ is realized also by value $\nabla''^{\beta}_{SE(basic)}$.

We introduce BIG (B) and SMALL (S) fuzzy sets to estimate the noise presence in a central pixel for each a sliding window. A big membership degree (≈1) in the SMALL set shows that the central pixel is free of noise, and a large membership degree in the BIG set shows that the central pixel is noisy one with large probability. To calculate membership degrees for fuzzy gradient and fuzzy vectorial values, we use the following Gaussian membership functions:

$$\mu(F^{\beta}_{\gamma}\text{SMALL}) = \begin{cases} 1, & F^{\beta}_{\gamma} < med_F \\ \exp\left\{-\left[(F^{\beta}_{\gamma} - med_F)^2 / 2\sigma^2_F\right]\right\}, & \text{otherwise} \end{cases}, \tag{3}$$

$$\mu(F^{\beta}_{\gamma}\text{BIG}) = \begin{cases} 1, & F^{\beta}_{\gamma} > med_F \\ \exp\left\{-\left[(F^{\beta}_{\gamma} - med_F)^2 / 2\sigma^2_F\right]\right\}, & \text{otherwise} \end{cases}, \tag{4}$$

where σ_1^2=1000, med_1=60 and med_2=10 for fuzzy gradient sets BIG and SMALL, respectively, σ_2^2=0.8, med_3=0.1 and med_4=0.615 for fuzzy angular deviation sets BIG and SMALL, respectively. The values med_3=0.01 and med_4=0.1 are changed in the case of use of 3D-FD filter. These values were found according to optimal PSNR and MAE values.

Table 1 presents the novel fuzzy rules that are based on gradient and angle variance values to determine if the central component is noisy or present local movement [3].

Table 1. Fuzzy rules used in the proposed 2D-FD and 3D-FD filters

Fuzzy Rule 1 introduces the membership level of $x^{\beta}_{(i,j)}$ in the set BIG for any γ direction: **IF** $(\nabla^{\beta}_{\gamma}\mathrm{B} \otimes \nabla^{\beta}_{\gamma(r_1)}\mathrm{S} \otimes \cdots \otimes \nabla^{\beta}_{\gamma(r_4)}\mathrm{B}) \otimes_1 (\theta^{\beta}_{\gamma}\mathrm{B} \otimes \theta^{\beta}_{\gamma(r_1)}\mathrm{S} \otimes \cdots \otimes \theta^{\beta}_{\gamma(r_4)}\mathrm{B})$, **THEN** the fuzzy gradient-vectorial value $\nabla^{\beta F}_{\gamma}\theta^{\beta F}_{\gamma}$ B.
Fuzzy Rule 2 presents the *noisy factor* gathering eight fuzzy gradient-directional values calculated for each a direction: **IF** $\nabla^{\beta F}_{N}\theta^{\beta F}_{N}\mathrm{B} \oplus \nabla^{\beta F}_{S}\theta^{\beta F}_{S}\mathrm{B} \oplus \cdots \oplus \nabla^{\beta F}_{SE}\theta^{\beta F}_{SE}\mathrm{B} \oplus \nabla^{\beta F}_{SW}\theta^{\beta F}_{SW}\mathrm{B}$ **THEN** the noisy factor r^{β} B.
Fuzzy Rule 3 characterizes the *movement and noise confidence* in a central pixel by neighbour fuzzy values in any γ by use the ***FIRST fuzzy gradient-vectorial difference*** *value* $\left(\nabla^{t\beta F}_{\gamma}\nabla^{tt\beta F}_{\gamma}\right)_{FIR}$: **IF** $(\nabla^{t\beta}_{\gamma}\mathrm{B} \otimes \nabla^{t\beta}_{\gamma(r_1)}\mathrm{S} \otimes \nabla^{t\beta}_{\gamma(r_2)}\mathrm{S} \otimes \nabla^{t\beta}_{\gamma(r_3)}\mathrm{B} \otimes \nabla^{t\beta}_{\gamma(r_4)}\mathrm{B}) \otimes_1 (\nabla^{tt\beta}_{\gamma}\mathrm{B} \otimes \nabla^{tt\beta}_{\gamma(r_1)}\mathrm{S} \otimes \nabla^{tt\beta}_{\gamma(r_2)}\mathrm{S} \otimes \nabla^{tt\beta}_{\gamma(r_3)}\mathrm{B} \otimes \nabla^{tt\beta}_{\gamma(r_4)}\mathrm{B})$, **THEN** $\left(\nabla^{t\beta F}_{\gamma}\nabla^{tt\beta F}_{\gamma}\right)_{FIR}$ B.
Fuzzy Rule 4 characterizes the *no movement confidence* in a central pixel in any γ direction, distinguishing different areas, such as, uniform region, edge or fine detail by use the ***SECOND fuzzy gradient-vectorial difference*** *value* $\left(\nabla^{t\beta F}_{\gamma}\nabla^{tt\beta F}_{\gamma}\right)_{SEC}$: **IF** $(\nabla^{t\beta}_{\gamma}\mathrm{S} \otimes \nabla^{t\beta}_{\gamma(r_1)}\mathrm{S} \otimes \nabla^{t\beta}_{\gamma(r_2)}\mathrm{S}) \oplus (\nabla^{tt\beta}_{\gamma}\mathrm{S} \otimes \nabla^{tt\beta}_{\gamma(r_1)}\mathrm{S} \otimes \nabla^{tt\beta}_{\gamma(r_2)}\mathrm{S})$, **THEN** $\left(\nabla^{t\beta F}_{\gamma}\nabla^{tt\beta F}_{\gamma}\right)_{SEC}$ S.
Fuzzy Rule 5 estimate the movement and noise level in a central component using the fuzzy values determined for all directions by use the ***fuzzy noisy factor*** r^{β}: **IF** $\left(\nabla^{t\beta F}_{SE}\nabla^{tt\beta F}_{SE}\right)_{FIR}\mathrm{B} \oplus \left(\nabla^{t\beta F}_{S}\nabla^{tt\beta F}_{S}\right)_{FIR}\mathrm{B} \oplus ,\ldots, \oplus \left(\nabla^{t\beta F}_{N}\nabla^{tt\beta F}_{N}\right)_{FIR}\mathrm{B}$, **THEN** r^{β} B.
Fuzzy Rule 6 defines the ***no movement confidence factor*** η^{β}: **IF** $\left(\nabla^{t\beta F}_{SE}\nabla^{tt\beta F}_{SE}\right)_{SEC}\mathrm{S} \oplus \left(\nabla^{t\beta F}_{S}\nabla^{tt\beta F}_{S}\right)_{SEC}\mathrm{S} \oplus ,\ldots, \oplus \left(\nabla^{t\beta F}_{N}\nabla^{tt\beta F}_{N}\right)_{SEC}\mathrm{S}$, **THEN** η^{β} S.
where $A \otimes B = A\,\mathrm{AND}\,B$, $A \otimes_1 B = \min(A,B)$, $A \oplus B = \max(A,B)$, and *P*B and *Q*S denote that value *P* is BIG and value *Q* is SMALL, respectively.

The parameters r^{β} and η^{β} can be effectively applied in the decision: if a central pixel component is noisy, or is in movement, or is a free one. Fuzzy Rules 1 and 2, and from 3 to 6 determine the 2D-FD and 3D-FD algorithm based on fuzzy parameters, respectively.

The noisy factor is used as a threshold to distinguish among a noisy pixel and a free noise one. If $r^{\beta} \geq 0.3$, the filtering procedure is applied employing the fuzzy gradient-vectorial values as weights. The fuzzy weights are used in the standard negator function $\varsigma(x) = 1 - x$, $x \in [0,1]$ defined as $\rho\left(\nabla^{\beta F}_{\gamma}\theta^{\beta F}_{\gamma}\right) = 1 - \nabla^{\beta F}_{\gamma}\theta^{\beta F}_{\gamma}$, this value origins a fuzzy membership value in a new fuzzy set defined as FREE (free noise). The fuzzy weight for central pixel in FREE fuzzy set is $\rho\left(\nabla^{\beta F}_{(0,0)}\theta^{\beta F}_{(0,0)}\right) = 3 \cdot \sqrt{1 - r^{\beta}}$. In opposite case, the output is presented as unchanged central pixel $y^{\beta}_{output} = x^{\beta(j)}_{\gamma}$.

To enhance the noise suppression capabilities of the proposed 3D filter, we use at the final stage the 2D-FD filter that permits to decrease the influence of the non-stationary noise left by temporal filter. Table 2 shows some modifications of the

Table 2. Parameters of 2D-FD Filter

2D-FD Filter	
Initial stage	Final stage
$r^{\beta} \geq 0.3$	$r^{\beta} \geq 0.5$
$\rho\left(\nabla^{\beta}\theta^{\beta}\right)=\left(\sum_{\dot{\gamma}}\rho\left(\nabla_{\dot{\gamma}}^{\beta F}\theta_{\dot{\gamma}}^{\beta F}\right)+3\cdot\sqrt{1-r^{\beta}}\right)/2$	$\rho\left(\nabla^{\beta}\theta^{\beta}\right)=\left(\sum_{\dot{\gamma}}\rho\left(\nabla_{\dot{\gamma}}^{\beta F}\theta_{\dot{\gamma}}^{\beta F}\right)+5\cdot\sqrt{1-r^{\beta}}\right)/2$
$\rho\left(\nabla_{(0,0)}^{\beta F}\theta_{(0,0)}^{\beta F}\right)=3\cdot\sqrt{1-r^{\beta}}$	$\rho\left(\nabla_{(0,0)}^{\beta F}\theta_{(0,0)}^{\beta F}\right)=5\cdot\sqrt{1-r^{\beta}}$
If condition $sum^{\beta} \geq \rho\left(\nabla^{\beta}\theta^{\beta}\right)$ until $\rho\left(\nabla_{\dot{\gamma}}^{\beta F}\theta_{\dot{\gamma}}^{\beta F}\right)^{(2)}$ is not satisfied, *total weight* is updated according to $\rho\left(\nabla^{\beta}\theta^{\beta}\right)=\left(\rho\left(\nabla^{\beta}\theta^{\beta}\right)-\rho\left(\nabla_{\dot{\gamma}}^{\beta F}\theta_{\dot{\gamma}}^{\beta F}\right)^{(1)}\right)/2\cdot$	

2D-FD applied after the 3D-FD that should be realized because of the non-stationary nature of noise according the best PSNR and MAE criteria results.

The output of 2D-FD filter is formed by selection of one of the neighbour pixels or central component. The condition $j \leq 2$ avoids the selection of the farther pixels, otherwise, the total weight is upgraded. Finally, the algorithm of 2D-FD filter is realized as follows:

1) Let calculate the fuzzy weights by use an ordering procedure: $x_{\gamma}^{\beta}=\left\{x_{SW}^{\beta},\ldots,x_{(i,j)}^{\beta},\ldots,x_{NE}^{\beta}\right\}$ and $x_{\gamma}^{\beta(1)} \leq x_{\gamma}^{\beta(2)} \leq \cdots \leq x_{\gamma}^{\beta(9)}$ implies that the fuzzy weights are given by $\rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(1)} \leq \rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(2)} \leq \cdots \leq \rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(9)}$, where $\dot{\gamma}=\{N,E,S,W,(i,j),NW,NE,SE,SW\}$, permitting to remove the values more outlying from the central pixel (*i,j*).
2) We define $sum^{\beta}+=\rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(9)}$ with *j*=9, decreasing *j* from 9 to 1 that is valid until the condition $sum^{\beta} \geq \rho\left(\nabla^{\beta}\theta^{\beta}\right)$ can be satisfied (where $\rho\left(\nabla^{\beta}\theta^{\beta}\right)=\left(\sum_{\gamma}\rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)+3\sqrt{1-r^{\beta}}\right)/2$). When *j* satisfies this condition, the *j*th ordered value $x_{\gamma}^{\beta(j)}=y_{output}^{\beta}$ is selected as the output filtered value. If $j \leq 2$, the next step of algorithm is realized.
3) If $j \leq 2$, it should be computed the weights $\rho_{j\leq 2}\left(\nabla^{\beta}\theta^{\beta}\right)=\left(\rho\left(\nabla^{\beta}\theta^{\beta}\right)-\rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(1)}-\rho\left(\nabla_{\gamma}^{\beta F}\theta_{\gamma}^{\beta F}\right)^{(2)}\right)/2$, defining $sum^{t\beta}+=\rho\left(\nabla_{\gamma}^{\beta(j)}\theta_{\gamma}^{\beta(j)}\right)$, and restore *j*=9, decreasing *j* from 9 to 1 that is valid until the condition $sum^{t\beta} \geq \rho\left(\nabla^{\beta}\theta^{\beta}\right)$. When *j* satisfies this condition, the *j*th ordered value $x_{\gamma}^{\beta(j)}=y_{output}^{\beta}$ is selected as the output filtered value.

Figure 2 exposes the block diagram of the proposed 3D-FD filter, where the *j-th* component pixel should be chosen, if it satisfies the proposed conditions, guaranteeing edges and fine detail preservation according to ordering criterion in the selection of the nearest pixels to the central one in *t*-1 and *t* frames.

3 Experimental Results

The described 3D-FD filter has been evaluated, and its performance has been compared with different filters proposed in literature [1-6, 10-13].

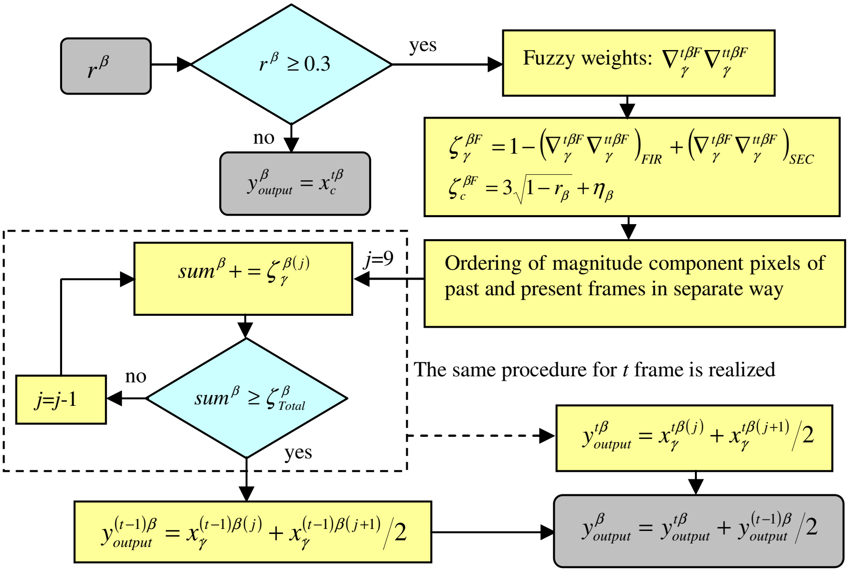

r^{β}
$r^{\beta} \geq 0.3$
yes
Fuzzy weights: $\nabla_{\gamma}^{t\beta F}\nabla_{\gamma}^{tt\beta F}$
no
$y_{output}^{\beta} = x_{c}^{t\beta}$
$\zeta_{\gamma}^{\beta F} = 1-\left(\nabla_{\gamma}^{t\beta F}\nabla_{\gamma}^{tt\beta F}\right)_{FIR} + \left(\nabla_{\gamma}^{t\beta F}\nabla_{\gamma}^{tt\beta F}\right)_{SEC}$
$\zeta_{c}^{\beta F} = 3\sqrt{1-r_{\beta}} + \eta_{\beta}$
$sum^{\beta} += \zeta_{\gamma}^{\beta(j)}$
j=9
Ordering of magnitude component pixels of past and present frames in separate way
no
j=j-1
$sum^{\beta} \geq \zeta_{Total}^{\beta}$
yes
The same procedure for t frame is realized
$y_{output}^{t\beta} = x_{\gamma}^{t\beta(j)} + x_{\gamma}^{t\beta(j+1)}/2$
$y_{output}^{(t-1)\beta} = x_{\gamma}^{(t-1)\beta(j)} + x_{\gamma}^{(t-1)\beta(j+1)}/2$
$y_{output}^{\beta} = y_{output}^{t\beta} + y_{output}^{(t-1)\beta}/2$

Fig. 2. Block diagram of proposed 3D Fuzzy Directional (3D-FD) algorithm

The performance criteria used to compare the restoration performance of various filters were the *peak signal-to-noise ratio* (PSNR) for evaluation of noise suppression, the *mean absolute error* (MAE) for quantification of edges and detail preservation, the *mean chromaticity error* (MCRE) for evaluation of chromaticity retention, and the *normalized color difference* (NCD) for quantification of color perceptual error [1-6].

The 176x144 QCIF video colour sequence "Miss America" was contaminated artificially by 5 and 15% of impulsive noise in independent way for each a channel.

Table 3 shows the performance results in the case of averaging of 100 frames of video sequence "Miss America". In the case of 5% of degradation, the proposed 3D-FD filter provides the best restoration performance, and for 15% of impulsive noise, one can see that the best PSNR performance is given by 3D-AVDATM filter but the best detail preservation and chromaticity properties criteria are realized by proposed 3D-FD filter. Figure 3 depicts the zoomed filtered frames of video sequence "Miss America" in the case of 15% of impulsive noise, where the proposed 3D-FD filter preserves better preservations of edges and fine details, and chromaticity properties against other filters.

To provide better evaluation capabilities, we have implemented some promising algorithms on DSP to a Real-Time evaluation with real video sequences. So, this can provide reliability of the proposed filter against some better algorithms found in scientific literature. Table 4 presents the processing time values in filtering of 20 video frames for several filters using the Texas Instruments EVM DM642. From these results, we observe that the proposed 3D-FD filter presents good capabilities in the real time environment.

Table 3. Performance results in the video sequence “Miss America” **using** different filters

Filters	5% of impulsive noise				15% of impulsive noise			
	PSNR	MAE	NCD	MCRE	PSNR	MAE	NCD	MCRE
3D-FD	**39.59**	**0.372**	**0.002**	**0.003**	34.33	**1.180**	**0.005**	**0.008**
3D-MF	35.12	2.514	0.009	0.013	34.36	2.701	0.010	0.014
3D-VMF	34.86	2.544	0.009	0.013	34.18	2.708	0.010	0.014
3D-VDKNNVM[10]	33.48	3.106	0.011	0.015	32.37	3.428	0.012	0.016
3D-GVDF [2]	33.76	2.905	0.011	0.014	33.70	2.847	0.010	0.014
3D-AVDATM [11]	36.97	1.112	0.004	0.006	**35.38**	1.714	0.006	0.009
3D-ATM [12,13]	35.22	2.569	0.009	0.013	34.42	2.767	0.010	0.013
3D-KNNF [12,13]	37.21	1.909	0.007	0.010	30.09	3.838	0.014	0.020

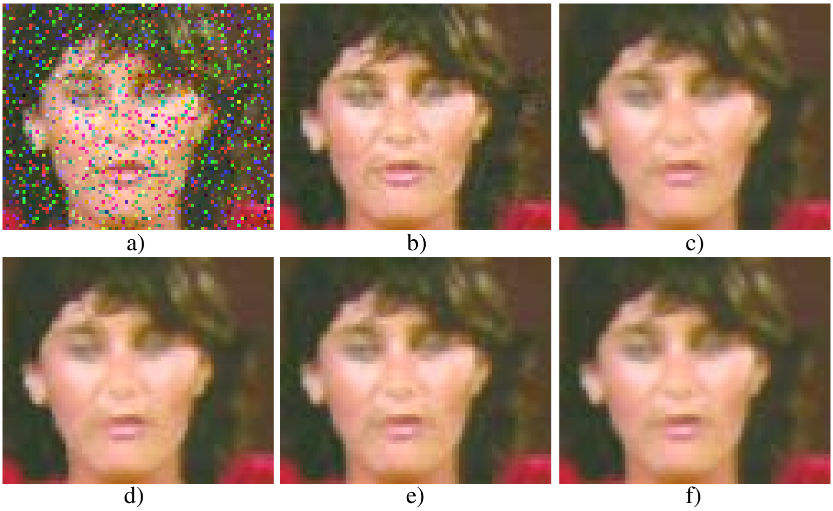


Fig. 3. Subjective visual qualities of restored 10th frame of “Miss America” video sequence, a) Zoomed image region contaminated by **impulsive noise of 15% intensity**, b) Designed 3D-FD, c) 3D-VMF, d) 3D-GVDF, e) 3D-AVDATM, f) 3D-ATM

Table 4. Time values needed for processing 20 frames of video sequence “Miss America”

Filters	Processing time in seconds		
	Maximum	Average	Total
3D-FD	**7.533**	**7.440**	**148.806**
3D-MF	0.0065	0.0057	0.114
3D-VMF	0.075	0.075	1.496
3D-GVDF [2]	28.52	25.6	512.02
3D-ATM	0.1347	0.134	2.681
3D-AVDATM	25.551	24.867	497.356
3D-KNNF [12,13]	0.103	0.102	2.04

4 Conclusions

The proposed 3D-FD filter uses fuzzy set theory and order vector statistic technique to provide better performance in noise suppression, edge and fine detail preservation, and chromaticity characteristics for video colour sequence denoising in comparison with existed filtering approaches in terms of objective criteria, as well subjective perception by human viewer. Finally, the Real-Time evaluation was realized using EVM DM642 presenting good capabilities in the real-time environment.

Acknowledgments. This work is supported by National Polytechnic Institute of Mexico and CONACyT (project 81599).

References

1. Schulte, S., Morillas, S., Gregori, V., Kerre, E.: A new fuzzy color correlated impulse noise reduction method. IEEE Trans. Image Process. 16(10), 2565–2575 (2007)
2. Plataniotis, K.N., Androutsos, D., Vinayagamoorthy, S., Venetsanopoulos, A.N.: Color image processing using adaptive multichannel filters. IEEE Trans. Image Process. 6(7), 933–949 (1997)
3. Ponomaryov, V.I., Gallegos-Funes, F.J., Rosales-Silva, A.: Real-time color imaging based on RM-Filters for impulsive noise reduction. J. Imaging Science and Technology 49(3), 205–219 (2005)
4. Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopoulos, A.N.: Selection weighted vector directional filters. Comput. Vision and Image Underst. 94, 140–167 (2004)
5. Smolka, B., Lukac, R., Chydzinski, A., Plataniotis, K.N., Wojciechowski, W.: Fast adaptive similarity based impulsive noise reduction filter. Real-Time Imag., 261–276 (2003)
6. Lukac, R.: Adaptive vector median filtering. Pattern Recognition Letters 24, 1889–1899 (2003)
7. Yin, H.B., Fang, X.Z., Wei, Z., Yang, X.K.: An improved motion-compensated 3-D LLMMSE filter with spatio–temporal adaptive filtering support. IEEE Trans. Circuits Syst. Video Techn. 17(12), 1714–1727 (2007)
8. Ghazal, M., Amer, A., Ghrayeb, A.: A Real-Time Technique for Spatio–Temporal Video Noise Estimation. IEEE Trans. Circuits Syst. Video Techn. 17(12), 1690–1699 (2007)
9. Mélange, T., Nachtegael, M., Kerre, E.E., Zlokolica, V., Schulte, S., De Witte, V., Pižurica, A., Philips, W.: Video denoising by fuzzy motion and details adaptive averaging. J. Electron. Imag. 17 0430051-19 (2008)
10. Ponomaryov, V.: Real-time 2D-3D filtering using order statistics based algorithms. J. Real-Time Image Process. 1(3), 173–194 (2007)
11. Ponomaryov, V., Rosales, A., Gallegos, F., Loboda, I.: Adaptive vector directional filters to process multichannel images. IEICE Trans. Funds. Electronics Comms. Computer Sciences, E90-B 2, 429–430 (2007)
12. Zlokolica, V., Philips, W., Van De Ville, D.: A new non-linear filter for video processing. In: Proc. Third IEEE Benelux Signal Processing Symposium (SPS 2002), Leuven, Belgium, March 2002, pp. 221–224 (2002)
13. Zlokolica, V., Schulte, S., Pizurica, A., Philips, W., Kerre, E.: Fuzzy logic recursive motion detection and denoising of video sequences. J. Electron. Imag. 15(2), 023008 (2006)

Image Authentication Scheme Based on Self-embedding Watermarking

Clara Cruz-Ramos, Rogelio Reyes-Reyes, Mariko Nakano-Miyatake,
and Héctor Pérez-Meana

ESIME Culhuacan, National Polytechnic Institute of México
Av. Santa Ana No. 1000, Col. San Francisco Culhuacan, México D.F., México
ccruza@ipn.mx, mariko@infinitum.com.mx

Abstract. This paper presents a block-wise and content-based semi-fragile image watermarking authentication scheme with location and recovery capability. Firstly the image is segmented by two regions: Region of Interest (ROI) and Region of Embedding (ROE). The watermark sequence is extracted from ROI and it is embedded into the middle frequency band of DCT coefficients of ROE. In the authentication stage, two watermark sequences extracted ROI and ROE, respectively, are used. If difference between both sequences of a ROI block is smaller than the predefined threshold value, the ROI block is determined authentic, otherwise the block is considered as tampered and it is recovered by the recovery process. The proposed scheme is evaluated from several points of view: watermark imperceptibility, capability of tamper detection, image quality of recovered regions and robustness of no-intentional manipulations, such as JPEG compression. The simulation results show fairly good performance of the proposed scheme.

Keywords: Image authentication, Semi-fragile watermarking, Self-embedding watermarking, Tamper detection, Tamper localization.

1 Introduction

With the growth of Internet, digital images play an important role to show some evidences in the news and reports in the digital media. Digital images captured by remote sensing technique provide us important information in several fields. However, using some software tools, digital images can be modified easily without any traces, causing economic and social damages. Therefore development of a reliable digital image authentication scheme is an urgent issue. Among several approaches, a watermarking based approach is considered as one of alternative solutions. In general, image authentication schemes can be classified into two approaches: signature-based authenticators [1,2] and watermarking-based authenticators [3-6]. The major difference between these two approaches is that the authentication message is embedded into the same digital media in watermarking-based authenticators, while it is transmitted or saved separately from digital media in the signature-based authenticators. In watermarking-based authenticators, they can be further classified into two schemes: fragile (complete authentication) [3, 4] or semi-fragile (content authentication) [5, 6] watermarking authentication schemes.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1005–1012, 2009.

Many watermarking based methods determine if the image is tampered or not, and some of them can localize the tampered regions [3-6], however, very few schemes have capability to recover the tampered region without original image [7, 8]. In this paper, an image authentication scheme, with a capability of tampered region localization and recovery, is proposed. In the proposed scheme, an image is segmented by two regions: Regions of Interest (ROI) and Regions of Embedding (ROE). ROI is a region which contains important information and it is required protection, for example regions of faces of persons involved in some scandal scene, while ROE is rest of the whole image after subtracting region belonged to ROI. ROE can be background of the image. The information of ROI is encoded to generate watermark sequence, and it is embedded into ROE of the same image in an imperceptible manner. In the authentication stage, two watermark sequences, extracted from ROI and ROE respectively, are used. If some blocks of ROI are detected as tampered, the recovery process performs to construct these blocks from the watermarked sequence extracted from ROE.

The proposed scheme is evaluated from several points of view: watermark imperceptibility in the watermarked image, capability of tampered regions detection, recovery capability of the altered region, and watermark robustness against no-intentional modification, such as JPEG compression. Simulation results show a fairy good performance about above three issues.

The rest of the paper is organized as follows. In Section 2, the proposed authentication method is described, and in Section 3 the experimental results are provided. Finally a conclusion of this paper is described in Section 4.

2 The Proposed Authentication Method

2.1 Watermark Sequence Generation

Generally in a photo image, some objects or some regions contain information more important than other regions. For example in an image of traffic accident, perhaps the regions of license plates of vehicles involved with the accident are more important than its background or other vehicles no related with the event. Therefore we define two regions in the image: region of interest (ROI) and region of embedding (ROE).

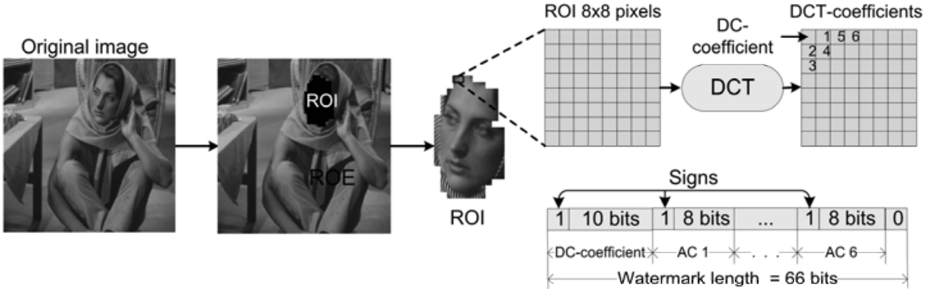


Fig. 1. Watermark sequence generation stage

ROI is important region of the image that requires a protection against malicious modification, while ROE is the rest of the image that no requires any protection. In the proposed algorithm, information of ROI is extracted to generate a watermark sequence and this sequence is embedded into ROE. Figure 1 illustrates the proposed watermark generation process that can be summarized as follows:

1) Subtract 127 from gray levels of the original image to force pixel values to be [-127,128]. It reduces DC-coefficient value after the image is transformed by DCT.
2) In the original image X, ROI is selected by owner and automatically ROE is determined in order that the following condition is satisfied.

$$ROI \cap ROE = \emptyset \ and \ \ ROI \cup ROE \,. \tag{1}$$

3) ROI region is divided into non-overlapping blocks of 8×8 pixels.
4) In each block of ROI, 66 bits watermark sequence is extracted as a following manner.
   a) Compute the 2D-DCT.
   b) The DC-coefficient is rounded and represented by 11 bits (10 bits for absolute value and 1 bit for sign). Because the maximum values of DC for 8x8 block of an image with range [-127,128] is 1016, it can be represented in a binary form using 11 bits, including sign bit.
   c) Encode each one of the first 6 lowest AC-coefficients, taking first 6 AC coefficients in the zig-zag order of the block, to 8 bits together with 1 sign bit (total 9 bits).
5) The length of watermark sequence of each ROI block is 66 bits, composed by 11 bits of DC-coefficient, 54 bits corresponded to the 6 AC-coefficients of DCT coefficients and finally we add 1 zero, which can be divided into 6 segments with 11 bits sequence per segment.

2.2 Watermark Embedding

The proposed watermark embedding process can be summarized as follows:

1) Using a user's key K, the mapping list between ROI blocks and ROE blocks is constructed.
2) Using this mapping list, each ROI block of 8x8 pixels is mapped into 6 ROE blocks, which are used to embed watermark sequence extracted from the ROI block.
3) In each selected 6 ROE blocks, following processes are carried out.

   a) Apply 2D-DCT to 6 ROE blocks.
   b) Quantify by a quantized matrix Q that corresponds to quality factor 70. This value is selected considering tradeoff between watermarked image quality and watermark robustness against JPEG compression. Quantization of DCT coefficients by Q is given by (2).

$$\tilde{C}(u,v) = \lfloor C(u,v)/Q(u,v) \rfloor\,. \tag{2}$$

Where $\mathrm{C(u,v)}$ and $\tilde{\mathrm{C}}\mathrm{(u,v)}$ are the $\mathrm{(u,v)}$-th DCT coefficient and it quantized version, respectively, $\lfloor \mathrm{x} \rfloor$ is lower nearest integer value of x.

c) Each 11 bits of watermark sequence is embedded into the LSB of the 11 DCT-coefficients of the middle frequency band of the selected 6 ROE blocks.
d) The watermarked DCT blocks are multiplied by Q.
e) It is transformed by the inverse DCT to get watermarked blocks.

4) Concatenating all watermarked blocks, the watermarked image is generated.

2.3 Authentication and Recovery

The authentication procedure verifies if the contents of the received image are authentic or not. To authenticate the image, two watermarks must be extracted and then compared. This authentication and recovery process are described as follows:

1) The first watermark $\mathrm{W_{ROIext}}$ is generated from the ROI blocks; these operations are same as the watermark generation process before described.
2) The second watermark $\mathrm{W_{ROEext}}$ is extracted from the ROE blocks. Using the same secret key to construct ROI-ROE mapping lists, the 6 corresponded ROE blocks are determined for each ROI block, from which $\mathrm{W_{ROEext}}$ is extracted.
3) For selected 6 ROE blocks, the following operations are carried out to get $\mathrm{W_{ROEext}}$

   a) Apply 2D-DCT to each one of 6 ROE blocks.
   b) DCT blocks are quantized by quantification matrix Q.
   c) 11 bits sequence is extracted from LSB of 11 AC coefficients in the middle frequency band of each ROE block.
   d) Concatenated 6 extracted sequences of longitude 11 bits to generate 66 bits $\mathrm{W_{ROEext}}$.
4) In the watermark comparison between $\mathrm{W_{ROIext}}$ and $\mathrm{W_{ROEext}}$, the tolerant threshold *Th* is employed to distinguish a content preserving operation from malicious manipulation. This authenticity check is given by (3).

$$\begin{array}{lll} if & \sum XOR(W_{ROIext}, W_{ROEext}) < Th & then\ the\ block\ is\ authentic \\ if & \sum XOR(W_{ROIext}, W_{ROEext}) \geq Th & then\ the\ block\ is\ modified \end{array} \tag{3}$$

Once the authenticity check indicates that a ROI block was tampered, the recovery process of this ROI block is triggered. The recovery process can be summarized as follows:

1) From the extracted watermark sequence $\mathrm{W_{ROEext}}$, last bit is eliminated to get a watermark sequence with 65 bits.
2) Assign the first 11 bits of $\mathrm{W_{ROEext}}$ to DC-component and the rest 54 bits are divided into 6 sequences with 9 bits and these are assigned to 6 lowest AC-coefficients of a recovery DCT block.

3) Compute the inverse 2D-IDCT of recovery DCT block to get a recovery block.
4) Replace the tampered ROI block by the recovery block.

3 Experimental Results

We conduct three experiments to evaluate performance of the proposed algorithm. The first experiment is to assess watermark imperceptibility, and in the second one the tamper detection and the recovery capability of the proposed algorithm are evaluated. Finally, in the third experiment, the watermark robustness to incidental modification such as JPEG compression is evaluated. In table 1, the values of some factors used during the evaluation are given.

Table 1. Parameter's values used during the evaluation process

Number of test images	256-gray level (8 bits/pixel)	100
Length of watermark sequence for each ROI block	W	66 bits
Threshold value used in (3)	Th	13
Number of ROI blocks used	[min, max]	[117,453]

Experimental results of three evaluations are described in the following sections.

3.1 Watermark Imperceptibility

Gray-scale images are used for these experiments. Figure 2 shows watermark imperceptibility using two images "Car" and "Camera" as examples. Figs. 2(a) and 2(d) show original image and figs. 2(b) and 2(e) show the watermarked images, respectively. Peak signal to noise ratio (PSNR) between the original image and the watermarked one are 36.8 dB and 33.17 dB, respectively. These results indicate that the image distortions incurred by the watermark embedding process are not significant. Also, in perceptual comparison by human visual system between the original images and the watermarked one, it is difficult to distinguish the difference between both images.

The watermark length of the cover image depends directly on the number of ROI blocks selected by the owner, and also watermarked image distortion depends on the embedded watermarked length. Figs. 2(c) and 2(f) show an example of possible ROI blocks selected by the owner, which are represented by black squares. Here 4% of blocks in Fig. 2(c) and 14.8% of blocks in Fig 2(f) are selected as ROI blocks. Fig. 3 shows a plot of the PSNR as a function with number of ROI blocks. In this figure, watermarked image quality (PSNR) is inversely proportioned by number of ROI blocks. The number of ROI blocks must satisfy (4), because each ROI block requires 6 ROE blocks.

$$Number(ROI) < {}^{NB(I)}/_{7} . \qquad (4)$$

where *NB(I)* is total number of blocks (8x8 pixels) of the Image I.

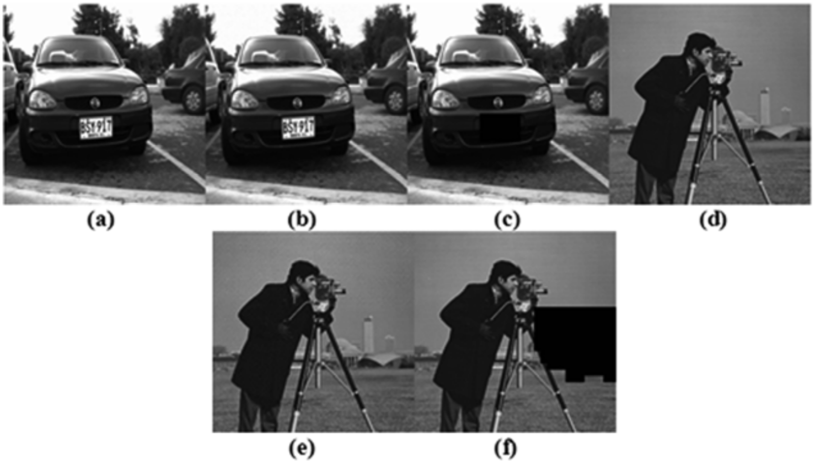


Fig. 2. Watermark imperceptibility, (a,d) Original images, (b,e) Watermarked images, (c,f) ROI blocks indicated by black blocks, which are assigned by owner of the image

3.2 Tamper Detection and Recovery Capability

To evaluate the tamper detection and recovery of the proposed authentication scheme, the watermarked images were tampered as shown by fig. 4(b) and (f). In the fig. 4 (b), number ‘7’ of license plate is tampered, modifying ‘9’ and in the fig. 4 (f), a tower behind cameraman is eliminated. As shown by fig. 2(c) and 2(f), tamped regions are assigned as ROI blocks. The tamper detection results are shown by fig. 4(c) and (g), where tampered ROI blocks are represented by black squares, and fig. 4(d) and (h) show images that the tampered regions were recovered.

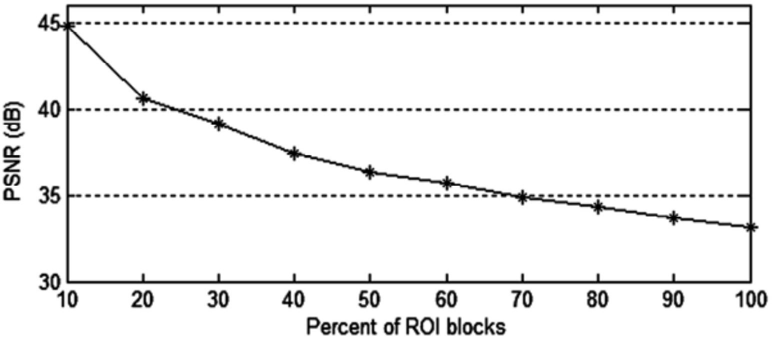


Fig. 3. Relationship between number of ROI blocks and watermarked image distortion

3.3 Watermark Robustness

Generally any images, including watermarked image, suffer some no-intentional modifications, such as compression or noise contamination, therefore, watermark robustness against these incidental modifications must be taken into account. In the

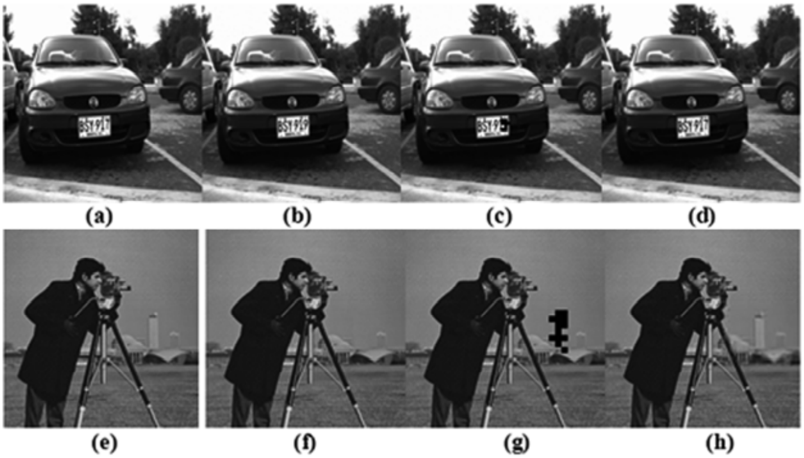


Fig. 4. (a,e) are watermarked images, (b,f) are tampered images, (c,g) show detection of tampered regions and (d,h) are recovered images

proposed authentication method, ROI information is embedded as watermark sequence into quantized DCT coefficients, which is generated by a predefined JPEG quality factor. This embedding domain guarantees that watermark sequence can be extracted in almost intact manner, after watermarked image suffer JPEG compression with a better quality factor than the predefined one. Therefore in the proposed scheme, embedded watermark sequence is robust to JPEG compression with a quality factor better than 70.

4 Conclusions

In this paper, a block-based image authentication with tamper detection and recovery capability is proposed. Firstly image is segmented by two regions: Regions of Interest (ROI) and Regions of Embedding (ROE). The watermark sequence is a compressed version of each ROI block and it is encoded a binary sequences with 66 bits. Then the 66 bits watermark sequence of each ROI block is embedded into corresponded 6 ROE blocks in its DCT domain. In the authentication stage, watermark sequence extracted from ROI blocks of the image under analysis is compared with the watermark sequences extracted from ROE blocks. If some ROI blocks are determined as tampered blocks, the recovery processes is triggered, in which, the tampered ROI blocks are recovered from the watermark sequence extracted from ROE blocks. Computer simulation results show fairly good performance of the proposed scheme, analyzing watermark imperceptibility, tamper detection and recovery capability and watermark robustness against no intentional attacks, such as JPEG compression. In the proposed scheme, recovered image of the tampered ROI blocks are sufficiently clear after watermarked image is compressed by JPEG compression with a reasonable quality factor (better than 70).

Acknowledgments. We thank to the National Polytechnic Institute of México and National Council for Science and Technology (CONACYT) of México for the support provided during the realization of this research.

References

1. Lu, C.-S., Liao, H.-Y.: Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme. IEEE Trans. Multimedia 5(2), 161–173 (2003)
2. Lou, D.-C., Ju., J.-L.: Fault Resilient and Compression Tolerant Digital Signature for Image authentication. IEEE Trans. Consumer Electron. 46(1), 31–39 (2000)
3. Maeno, K., Sun, Q., Chang, S.-F., Suto, M.: New Semi-Fragile Image Authentication Watermarking Techniques Using Random Bias and Nonuniform Quantization. IEEE Trans. Multimedia 8(1), 32–45 (2000)
4. Wong, P.-W., Memon, N.: Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification. IEEE Trans. Image Processing 10(10), 1593–1601 (2001)
5. Lin, C.-Y., Chang, S.-F.: A Robust Image Authentication Method Distinguishing JPEG compression from Malicious Manipulation. IEEE Trans. Circuit Syst. Video Technol. 11(2), 153–168 (2001)
6. Lu, Z.-M., Xu, D.-G., Sun, S.-H.: Multipurpose Image Watermarking Algorithm Based on Multistage Vector Quantization. IEEE Trans. Image processing 14, 822–831 (2005)
7. Lin, P.-L., Huang, P.-W., Peng, A.-W.: A Fragile Watermarking Scheme for Image Authentication with Localization and Recovery. In: Proc. of the IEEE sixth Int. Symp. on Multimedia Software Engineering, pp. 146–153 (2004)
8. Tsai, P., Hu, Y.-C.: A Watermarking-Based Authentication with Malicious Detection and Recovery. In: Int. Conf. of Information, Communication and Signal Processing, pp. 865–869 (2005)

Unified Experiment Design, Bayesian Minimum Risk and Convex Projection Regularization Method for Enhanced Remote Sensing Imaging

Yuriy Shkvarko, Jose Tuxpan, and Stewart Santos

Department of Electrical Engineering, CINVESTAV-IPN, Guadalajara, Mexico
{shkvarko,jtuxpan,ssantos}@gdl.cinvestav.mx

Abstract. We address new approach for enhanced multi-sensor imaging in uncertain remote sensing (RS) operational scenarios. Our approach is based on incorporating the projections onto convex solution sets (POCS) into the descriptive experiment design regularization (DEDR) and fused Bayesian regularization (FBR) methods to enhance the robustness and convergence of the overall unified DEDR/FBR-POCS procedure for enhanced RS imaging. Computer simulation examples are reported to illustrate the efficiency and improved operational performances of the proposed unified DEDR/FBR-POCS imaging techniques in the extremely uncertain RS operational scenarios.

Keywords: Convex sets, descriptive regularization, experiment design, multi-sensor imaging, remote sensing.

1 Introduction

In this study, we propose a unification of the previously developed descriptive experiment design regularization (DEDR) [1] and the fused Bayesian regularization (FBR) [2] methods for enhanced imaging in the remote sensing (RS) operational scenarios with model uncertainties. The operational uncertainties are associated with the unknown statistics of random perturbations of the signal formation operator (SFO) in the turbulent medium, imperfect sensor system calibration, finite dimensionality of measurements, multiplicative signal-dependent speckle noise, uncontrolled antenna vibrations and random carrier trajectory deviations in the case of SAR. The general DEDR method for solving such class of uncertain RS inverse problems has been constructed in our previous study [3] as an extension of the statistically optimal maximum likelihood (ML) technique [1], in which the spatial spectrum pattern (SSP) estimation error was minimized in a descriptively balanced fashion via weighted maximization of spatial resolution over minimization of resulting noise energy algorithmically coupled with the worst-case statistical performance optimization-based convex regularization. In this paper, we are focused on the design of the unified DEDR/FBR method employing the idea of incorporating the projections onto convex solution sets (POCS) into the corresponding DEDR/FBR-related solution operators to enforce the robustness and convergence. The crucial practical issue relates to proper

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1013–1020, 2009.

adjustment of the regularization parameters in the unified DEDR/FBR-POCS iterative reconstructive technique to the particular uncertain RS operational scenario. The advantage in using the developed method over the previously proposed RS imaging and de-speckling techniques is demonstrated through the reported computer simulation experiments performed using the elaborated virtual remote sensing laboratory (VRSL) software.

2 Problem Formalism

Referring to our previous studies [1]–[3], the random signal u at the output of the sensor system antenna (SAR system in this particular study) moved by the carrier along the deviated linear trajectory $\boldsymbol{\rho}(t)$ in the time instance t relates to the field e scattered from the probing surface through the integral equation of observation

$$u(\mathbf{p}) = (\tilde{\mathbf{S}}e(\mathbf{r}))(\mathbf{p}) + n(\mathbf{p}) = \int_R \tilde{S}(\mathbf{p},\mathbf{r})e(\mathbf{r})d\mathbf{r} + n(\mathbf{p}) \tag{1}$$

where $\mathbf{p} = (t, \boldsymbol{\rho}(t))$ defines the time-space trajectory points, the complex scattering function $e(\mathbf{r})$ represents the random scene reflectivity over the probing surface in the plane of the scanned scene [6]; $\mathbf{r}$ is a vector of the scan parameters, usually the polar, cylindrical or Cartesian coordinates of the probing surface; the uncertain SFO $\tilde{\mathbf{S}}$ is defined by the integral at the right hand of (1) with the nominal kernel $S(\mathbf{p},\mathbf{r}) = \left\langle \tilde{S}(\mathbf{p},\mathbf{r}) \right\rangle$ specified by the time-space modulation of signals employed in a particular imaging SAR system [4]. The variations about the mean $\delta S(\mathbf{p},\mathbf{r}) = \tilde{S}(\mathbf{p},\mathbf{r}) - S(\mathbf{p},\mathbf{r})$ pertain to the random perturbation component in the SFO.

The spatial spectrum pattern (SSP) $b(\mathbf{r}) = \langle |e(\mathbf{r})|^2 \rangle$ represents the ensemble average of the squared modulus of the random complex scene reflectivity $e(\mathbf{r})$ as a function over the analysis domain $R \ni \mathbf{r}$ and is referred to as a desired RS image to be reconstructed from the measurement data recordings. The vector-form model of (1) is given by discrete-form equation of observation (EO) [3]

$$\mathbf{u} = \tilde{\mathbf{S}}\mathbf{e} + \mathbf{n} = \mathbf{S}\mathbf{e} + \boldsymbol{\Delta}\mathbf{e} + \mathbf{n}, \tag{2}$$

where $\mathbf{u}$, $\mathbf{n}$ and $\mathbf{e}$ define the vectors composed of the coefficients $\{u_m\}$, $\{n_m\}$ and $\{e_k\}$ of the discrete-form approximations of the fields u, n and e with respect to the selected orthogonal decomposition function set $\{h_m(\mathbf{p})\}$ in the observation domain and the pixel set $\{g_k(\mathbf{r})\}$ in the scene domain, respectively [3]. The matrix-form representation of the uncertain SFO in (2) was formalized in [3] by

$$\tilde{\mathbf{S}} = \mathbf{S} + \boldsymbol{\Delta}\,. \tag{3}$$

The $M\times K$ nominal SFO matrix $\mathbf{S}$ in (2), (3) is composed of the scalar products $\{ [Sg_k, h_m]_{\mathrm{U}} \}$ [1], while all problem model uncertainties are attributed to the distortion term $\boldsymbol{\Delta}$. We refer to our previous study [3], where the distortions in the random medium were explained based on the propagation theory models [6]. Note that in practice, one cannot attribute the exact portion of the composite SFO perturbation term $\boldsymbol{\Delta}$

to a particular source of disturbances, thus cannot separate in (3) the uncertainties caused by the turbulent medium effects, speckle noise or the observation mismatch errors as those are randomly mixed in the $\mathbf{\Delta}$. These practical aspects motivated our adopting in [3] the robust statistical treatment of the irregular SFO perturbations $\mathbf{\Delta}$ as a random zero-mean matrix with the bounded second-order moment, i.e.

$$\left\langle \mathbf{\Delta} \right\rangle = \mathbf{0}; \quad \left\langle \| \mathbf{\Delta} \|^2 \right\rangle = \langle \mathrm{tr}\{ \mathbf{\Delta}\mathbf{\Delta}^+ \}\rangle \le \eta \tag{4}$$

where $\| \mathbf{\Delta} \|^2 = \mathrm{tr}\{ \mathbf{\Delta}\mathbf{\Delta}^+ \}$ defines the squared matrix norm, $\mathrm{tr}\{\cdot\}$ is the trace operator, superscript $^+$ defines the Hermitian conjugate (conjugate transpose), and η is the bounding constant [3].

Because of an incoherent nature of the scattering function $e(\mathbf{r})$, vector $\mathbf{e}$ in the equation of observation (2) is characterized by a diagonal correlation matrix, $\mathbf{R_e} = \mathrm{diag}(\mathbf{b}) = \mathbf{D}(\mathbf{b})$, in which the K×1 vector $\mathbf{b}$ of the principal diagonal (composed of the elements $b_k = \left\langle | e_k |^2 \right\rangle$; $k = 1, \ldots, K$) is referred to as the vector-form SSP. The problem that we solved in our previous studies [1]–[3] was to derive an estimate $\hat{\mathbf{b}}$ of the SSP vector and to reconstruct the desired SSP distribution

$$\hat{b}_{(K)}(\mathbf{r}) = \sum\nolimits_{k=1}^{K} \hat{b}_k g_k(\mathbf{r}) \tag{5}$$

over the pixel-formatted observation scene $R \ni \mathbf{r}$ by processing the data vector $\mathbf{u}$ (in the operational scenario with the single processed uncertain data realization) or J>1 whatever available recorded independent realizations $\{ \mathbf{u}_{(j)}; j = 1, \ldots, J \}$ of the data (in the scenario with multiple observations) collected with a particular system operating in the uncertain RS environment.

3 Phenomenology

3.1 DEDR Method

To alleviate the ill-posedness of the SSP reconstruction problem (5) with the uncertain observation model (2)–(4), the DEDR method was constructed in [3] given by

$$\hat{\mathbf{b}}_{DEDR} = \{ \mathbf{F}_{DEDR}\mathbf{Y}\mathbf{F}^+_{DEDR} \}_{\mathrm{diag}} = \{\mathbf{K}\mathbf{S}^+\mathbf{R}_\Sigma^{-1}\mathbf{Y}\mathbf{R}_\Sigma^{-1}\mathbf{S}\mathbf{K}\}_{\mathrm{diag}} \tag{6}$$

that estimates the SSP vector $\hat{\mathbf{b}}$ via applying the DEDR-optimal solution operator

$$\mathbf{F}_{DEDR} = \mathbf{K}\mathbf{S}^+\mathbf{R}_\Sigma^{-1} \tag{7}$$

to the data matrix $\mathbf{Y}$ composed of the uncertain data measurements, i.e. the rank-1 (ill-conditioned) outer product matrix $\mathbf{Y} = \mathbf{Y}_{(rank\text{-}1)} = \mathbf{u}\mathbf{u}^+$ in the scenario with the single recorded data realization (e.g., single-look imaging SAR applications), and the rank-J empirical estimated correlation matrix $\mathbf{Y} = \mathbf{Y}_{(rank\text{-}J)} = \left(1/J\right)\sum\nolimits_{j=1}^{J} \mathbf{u}_{(j)}\mathbf{u}^+_{(j)}$ in the scenario with J>1 independent multiple observations [3].

The $\mathbf{S}^+$ in the solution operator (7) represents the adjoint (Hermitian conjugate [5]) to the nominal SFO matrix $\mathbf{S}$, and $\mathbf{R}_\Sigma^{-1}$ is the inverse of the augmented (diagonal loaded) noise correlation matrix defined by [3], $\mathbf{R}_\Sigma = \mathbf{R}_\Sigma(\beta) = (\mathbf{R_n} + \beta\,\mathbf{I})$. In the practical RS scenarios [4], [5], (and specifically, in the SAR imaging applications), it is a common practice to accept the robust white additive noise model, i.e. $\mathbf{R_n} = N_0\mathbf{I}$, attributing the unknown correlated noise component as well as the speckle to the composite uncertain noise term $\boldsymbol{\Delta}\mathbf{e}$ in (2), in which case $\mathbf{R}_\Sigma = N_\Sigma\mathbf{I}$, $N_\Sigma = N_0 + \beta$ with the composite noise variance $N_\Sigma = N_0 + \beta$, the initial N_0 augmented by the loading factor $\beta = \gamma\eta / \alpha \geq 0$ adjusted to the regularization parameter α, the Loewner ordering factor γ>0, and to the SFO uncertainty bound $\eta \geq \langle \mathrm{tr}\{\boldsymbol{\Delta\Delta}^+\}\rangle$ (see [3] for details).

Next, we refer to [3] for specifying the family of the DEDR-related estimators for the considered there feasible adjustments of the processing-level degrees of freedom $\{\alpha, N_\Sigma, \mathbf{A}\}$,

$$\hat{\mathbf{b}}^{(p)} = \{\mathbf{F}^{(p)}\mathbf{Y}\mathbf{F}^{(p)+}\}_{\mathrm{diag}}\,; \quad p = 1, \ldots, P, \tag{8}$$

where different employed solution operators $\{\mathbf{F}^{(p)};\ p = 1, \ldots, P\}$ specify the corresponding DEDR-related estimators.

3.2 FBR Method

The estimator that produces the high-resolution optimal (in the sense of the Bayesian minimum risk strategy) estimate $\hat{\mathbf{b}}$ of the SSP vector via processing the M-dimensional data recordings $\mathbf{u}$ applying the fused Bayesian-regularization (FBR) estimation strategy that incorporates nontrivial a priori geometrical and projection-type model information was developed in [1], [2]. The FBR method [1], [2] implies two-stage data processing. First, the vector of sufficient statistics (SS) is formed $\mathbf{v} = \{\mathbf{F}_{FBR}\mathbf{u}\mathbf{u}^+\mathbf{F}_{FBR}^+\}_{\mathrm{diag}}$ applying the regularized solution operator

$$\mathbf{F}_{FBR} = \mathbf{F}^{(6)} = (\mathbf{S}^+\mathbf{R}_\mathbf{n}^{-1}\mathbf{S} + \hat{\mathbf{D}}^{-1})^{-1}\mathbf{S}^+\mathbf{R}_\mathbf{n}^{-1} \tag{9}$$

to the sampled trajectory data signal $\mathbf{u}$. Second, the smoothing window $\mathbf{W}$ is applied to such SS to satisfy the regularizing consistency and metrics constraints [1], [2] that yields the resulting FBR estimator

$$\hat{\mathbf{b}}_{FBR} = \mathbf{W}\mathbf{v} = \mathbf{W}\{\mathbf{F}_{FBR}\mathbf{u}\mathbf{u}^+\mathbf{F}^+{}_{FBR}\}_{\mathrm{diag}}. \tag{10}$$

Thus, the FBR method may also be viewed as a particular member of the unified DEDR-related family (8), in which the additional pseudo averaging is performed applying the regularizing window $\mathbf{W}$.

4 POCS Regularized Unified DEDR/FBR Technique

To precede from the general-form DEDR and FBR estimators to the practically realizable SAR-adapted SSP reconstruction techniques, we follow the convex regularization paradigm invoked from the fundamental theorem of POCS [5]. Our approach incorporates the intrinsic factorization and sparseness properties of the SAR ambiguity functions [4], [7] into the construction of the POCS-regularized fixed-point iterative SSP reconstruction procedures that drastically reduces the overall computational load of the resulting algorithms.

To convert the general-from estimators (6) and (10) with the ML-optimally specified degrees of freedom [3] (i.e., $\alpha\mathbf{A} = \mathbf{D}(\hat{\mathbf{b}})$, $N_\Sigma = N_0+\beta$) to a unified POCS-regularized fixed-point iterative algorithm, we first, define a sequence of estimates $\{\hat{\mathbf{b}}_{[i]}\}$ as

$$\hat{\mathbf{b}}_{[i]} = \mathbf{P}\{\mathbf{K}_{[i]}\mathbf{S}^{+}\mathbf{YSK}_{[i]}\}_{\text{diag}} \tag{11}$$

$i = 0, 1, \ldots$, where $\mathbf{P}$ is a convergence enforcing projector (in our case, the POCS-regularizing operator) [5];

$$\mathbf{K}_{[i]} = \mathbf{K}(\hat{\mathbf{b}}_{[i]}) = (\boldsymbol{\Psi} + N_\Sigma \mathbf{D}^{-1}(\hat{\mathbf{b}}_{[i]}))^{-1} \tag{12}$$

represents the self-adjoint reconstruction operator at the ith iteration step and

$$\boldsymbol{\Psi} = \mathbf{S}^{+}\mathbf{S} \tag{13}$$

is the nominal system point spread function (PSF) operator [2]. Applying routinely the fixed-point technique [5] to the equation (12), we next, construct the unified POCS-regularized iterative SSP estimation algorithm

$$\hat{\mathbf{b}}_{[i+1]} = \mathbf{P}\,\hat{\mathbf{b}}_{[0]} + \mathbf{P}\,\mathbf{T}_{[i]}\hat{\mathbf{b}}_{[i]};\; i = 0, 1, \ldots \tag{14}$$

Here,

$$\mathbf{T}_{[i]} = \mathbf{T}_{[i]}(\hat{\mathbf{b}}_{[i]}) = 2\text{diag}(\{\boldsymbol{\Omega}_{[i]}(\hat{\mathbf{b}}_{[i]})\}_{\text{diag}}) - \mathbf{H}_{[i]}(\hat{\mathbf{b}}_{[i]});\; i = 0, 1, \ldots \tag{15}$$

represents the solution-dependent matrix-form iteration operator, where

$$\boldsymbol{\Omega}_{[i]} = \boldsymbol{\Omega}_{[i]}(\hat{\mathbf{b}}_{[i]}) = \mathbf{I} - \boldsymbol{\Psi} - N_\Sigma \mathbf{D}^{-1}(\hat{\mathbf{b}}_{[i]})\;; \tag{16}$$

$$\mathbf{H}_{[i]} = \mathbf{H}_{[i]}(\hat{\mathbf{b}}_{[i]}) = \boldsymbol{\Omega}_{[i]}(\hat{\mathbf{b}}_{[i]}) \circ \boldsymbol{\Omega}^{*}_{[i]}(\hat{\mathbf{b}}_{[i]})\;; \tag{17}$$

$\circ$ denotes the Shur-Hadamar (element-by-element) matrix product, and the zero-step iteration

$$\hat{\mathbf{b}}_{[0]} = \hat{\mathbf{b}}_{MSF} = \{\mathbf{S}^{+}\mathbf{YS}\}_{\text{diag}} \tag{18}$$

is formed as an outcome of the conventional matched spatial filtering (MSE) algorithm from the DEDR family (8) specified for the adjoint SFO solution operator $\mathbf{S}^{+}$. The principal advantage of the fixed-point procedure (14) relates to the exclusion of the solution-dependent operator inversions (12), which are now performed in an indirect iterative fashion. This transforms the computationally extremely intensive general-form procedures (6), (10) into the iterative technique (14) executable in (near) real computational time.

5 Simulations

Having established the unified POCS-regularized DEDR/FBR-related iterative technique (14) for SSP reconstruction, in this section, we present the results of comparative numerical simulations of six different SSP formation/reconstruction algorithms, in particular, the conventional MSF algorithm [1], the best existing adaptive Lee despeckling algorithm [7], the non-constrained robust spatial filtering (RSF) algorithm from [3], the constrained RSF algorithm from [3], the adaptive spatial filtering (ASF) algorithm from [2], and the proposed here POCS-regularized unified DEDR/FBR algorithm (14). We considered a SAR imaging system operating in a typical uncertain RS imaging scenario. The operational uncertainties were simulated via incorporating random perturbations into the regular SFO and contaminating the data with composite multiplicative and additive noise. In the simulation experiments that we report in this paper, the PSF of the fractional SAR system was modeled by of a Gaussian "bell" function in both directions of the 2-D scene (in particular, of 16 pixel width at 0.5 from its maximum for the 512-by-512 pixel-formatted scene). The composite multiplicative noise was simulated as a realization of the χ_2^2-distributed random variables with the pixel mean value assigned to the actual degraded scene image pixel that directly obeys the RS speckle model [4], [7]. Such signal-dependent multiplicative image noise dominates the additive noise component in the data in the sense that $N_\Sigma >> N_0$, hence the estimate $\hat{N}_\Sigma$ performed empirically using the local statistics method [7] was used to adjust the regularization degrees of freedom in the DEDR/FBR-POCS procedure (14). Two scenes (the artificially synthesized and borrowed from the real-world RS imagery [8]) were tested. These scenes are displayed in Figures 1(a) and 1(b), respectively. The qualitative simulation results for six different simulated image formation/reconstruction procedures for the first simulated scene are presented in Fig. 2 and for the second scene in Fig. 3, respectively, as specified in the figure captions. The advantage of the unified DEDR/FBR-POCS technique over the previously proposed conventional MSF, de-speckling without DEDR/FBR enhancement and non-adaptive RSF algorithms is evident from the reported simulations.

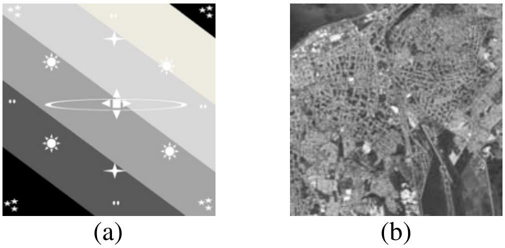


Fig. 1. Original test scenes: (a) artificially synthesyzed scene; (b) real-world RS scene borrowed from the high-resolution RS imagery [8]

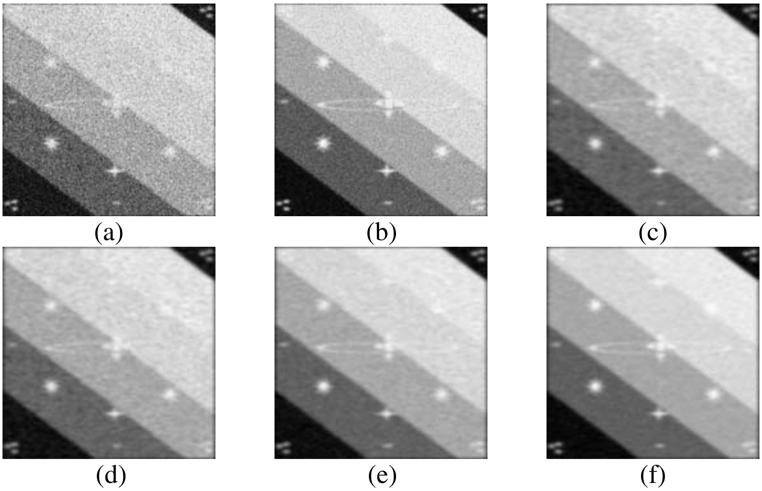


Fig. 2. Simulation results for the first test scene: (a) degraded SAR scene image formed applying the MSF method [1]; (b) adaptively de-speckled MSF image; (c) image reconstructed applying the non-constrained RSF algorithm [2]; (d) image reconstructed with the constrained RSF algorithm [3]; (e) image reconstructed applying the non-constrained ASF algorithm [2]; (f) image reconstructed applying the developed POCS-regularized DEDR/FBR method

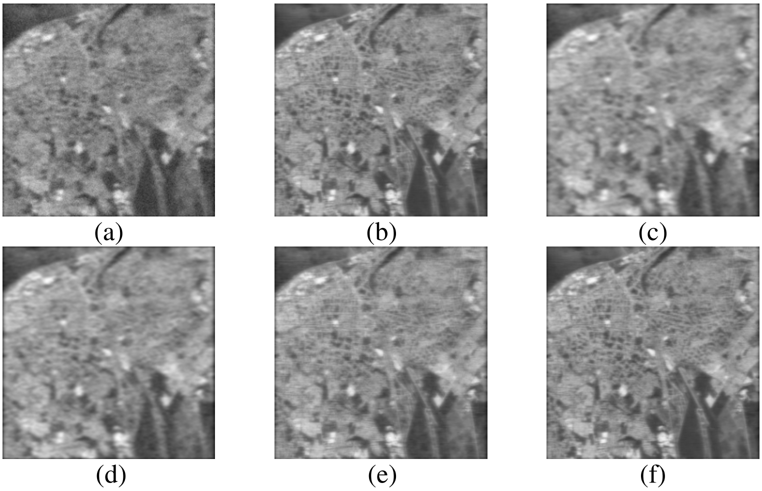


Fig. 3. Simulation results for the second test scene: (a) degraded SAR scene image formed applying the MSF method [1]; (b) adaptively de-speckled MSF image; (c) image reconstructed applying the non-constrained RSF algorithm [2]; (d) image reconstructed with the constrained RSF algorithm [3]; (e) image reconstructed applying the non-constrained ASF algorithm [3]; (f) image reconstructed applying the developed POCS-regularized DEDR/FBR method

6 Concluding Remarks

In this paper, we have presented the POCS-regularized fixed-point iterative DEDR/FBR method particularly adapted for enhanced RS imaging in the uncertain environment. The unified DEDR/FBR-POCS approach leads to the fixed-point SSP estimator that may be regarded as adaptive post-image-formation enhancement procedure. To facilitate it application for the uncertain imaging scenarios the adaptive scheme for evaluation of the operational degree of freedom (regularization parameter) directly from the uncertain RS data was incorporated into the SSP reconstruction algorithm. We have demonstrated that with such developed adaptive POCS-regularized DEDR/FBR technique, the overall RS image enhancement performances can be improved if compared with those obtained using conventional single-look SAR systems that employ auto-focusing techniques or the previously proposed adaptive despeckling and reconstruction filters that do not unify the POCS regularization with the DEDR/FBR method.

References

1. Shkvarko, Y.: Estimation of wavefield power distribution in the remotely sensed environment: Bayesian maximum entropy approach. IEEE Trans. Signal Proc. 50(9), 2333–2346 (2002)
2. Shkvarko, Y.: Unifying regularization and Bayesian estimation methods for enhanced imaging with remotely sensed data—Part I: Theory. IEEE Trans. Geoscience and Remote Sensing 42(5), 923–931 (2004)
3. Shkvarko, Y., Perez-Meana, H., Castillo-Atoche, A.: Enhanced radar imaging in uncertain environment: A descriptive experiment design regularization approach. Int. J. Navigation and Observation 2008, 1–11 (2008) Article ID 810816
4. Wehner, D.R.: High-Resolution Radar, 2nd edn. Artech House, Boston (1994)
5. Barrett, H.H., Myers, K.J.: Foundations of Image Science. Willey, New York (2004)
6. Ishimary, A.: Wave Propagation and Scattering in Random Media. IEEE Press, NY (1997)
7. Greco, M.S., Gini, F.: Statistical analysis of high-resolution SAR ground clutter data. IEEE Trans. Geoscience and Remote Sensing 45(3), 566–575 (2007)
8. Space Imaging, GeoEye Inc. (2009), `http://www.spaceimaging.com/quicklook`

Intelligent Experiment Design-Based Virtual Remote Sensing Laboratory

Yuriy Shkvarko, Stewart Santos, and Jose Tuxpan

Department of Electrical Engineering, CINVESTAV-IPN, Guadalajara, Mexico
{shkvarko,ssantos,jtuxpan}@gdl.cinvestav.mx

Abstract. We address unified intelligent descriptive experiment design regularization (DEDR) methodology for computer-aided investigation of new intelligent signal processing (SP) perspectives for collaborative remote sensing (RS) and distributed sensor network (SN) data acquisition, intelligent processing and information fusion. The sophisticated "Virtual RS Laboratory" (VRSL) software elaborated using the proposed DEDR methodology is presented. The VRLS provides the end-user with efficient computational tools to perform numerical simulations of different RS imaging problems. Computer simulation examples are reported to illustrate the usefulness of the elaborated VRSL for the algorithmic-level investigation of high-resolution image formation, enhancement, fusion and post-processing tasks performed with the artificial and real-world RS imagery.

Keywords: Computer simulations, experiment design, regularization, remote sensing, software.

1 Introduction

This paper is focused on the challenging problems of intelligent remote sensing (RS) data processing, distributed fusion, algorithm design and simulation software development. First, we address a unified intelligent descriptive experiment design regularization (DEDR) methodology for (near) real time formation/enhancement/reconstruction/post-processing of the RS imagery acquired with different types of sensors, in particular, conventional 2-D stationary arrays [1] and the mobile synthetic aperture radar (SAR) systems [2]. Second, we present the elaborated "Virtual Remote Sensing Laboratory" (VRSL) software that provides the end-user with efficient computational tools to perform numerical simulations of different collaborative RS imaging problems in various experiment design settings. The scientific challenge is to develop and investigate via the VRSL an intelligent signal processing (SP) perspective for collaborative RS data acquisition, adaptive processing and information fusion for the purposes of high-resolution RS imaging, search, discovery, discrimination, mapping and problem-oriented analysis of spatially distributed physical RS signature fields. The end-user oriented VRSL software is elaborated

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1021–1028, 2009.

directly to assist in system-level and algorithmic-level investigation of such multi-sensor collaborative image formation, enhancement, and post-processing tasks performed with the artificial and real-world RS imagery.

2 Unified DEDR Paradigm

The DEDR paradigm constitutes a methodology for solving a manifold of algorithm design problems related to high-resolution multisensor imaging and knowledge-based (KB) collaborative RS data processing. In the DEDR framework developed originally in [8], [10], [11] complex multisensor measurement data wavefields in the observation domain are modeled as operator transforms of the initial scene scattering fields degraded by clutter and noise. The formalism of such transforms is specified by the corresponding uncertain signal formation operator (SFO) models derived from scattering theory [2], [4]. In [8], [10], [11], we followed a generalized maximum entropy (ME) formalization of a priori information regarding the spatial spectrum patterns (SSPs) of the scattered wavefields that unify diverse RS imaging problem settings. Being nonlinear and solution-dependent, the optimal general-form DEDR estimators of the SSPs constructed in [8], [10] require computationally intensive adaptive signal processing operations that involve also the proper construction of the regularizing projections onto convex (solution) sets (POCS) ruled by the adopted fixed-point contractive iteration process. The fused KB DEDR algorithm design methodology [11] aggregates next the ME method with the diverse regularization and KB post-processing considerations. Such the methodology [11] enables one not only to form the atlas of the desired remote sensing signatures (RSS) extracted from the collaboratively processed multisensor RS imagery but also to perform their problem-oriented analysis in an intelligent KB fashion. Figure 1 presents the block-diagram of the addressed intelligent DEDR approach with the KB multisensor data fusion.

3 DEDR Phenomenology

Following [6], [8], [10] the DEDR is addressed to as a methodology that unifies the family of the previously developed nonparametric high-resolution RS imaging techniques. Such unified formalism allows involving into the DEDR different convex regularization and neural computation paradigms (e.g., the POCS regularization and KB method fusion) that enables the end user to modify the existing techniques via incorporation of some controllable algorithmic-level "degrees of freedom" as well as design a variety of efficient aggregated/fused data/image processing methods. The data/image processing tasks that may be performed applying the DEDR methodology can be mathematically formalized in the terms of the following unified optimization problem [8], [10].

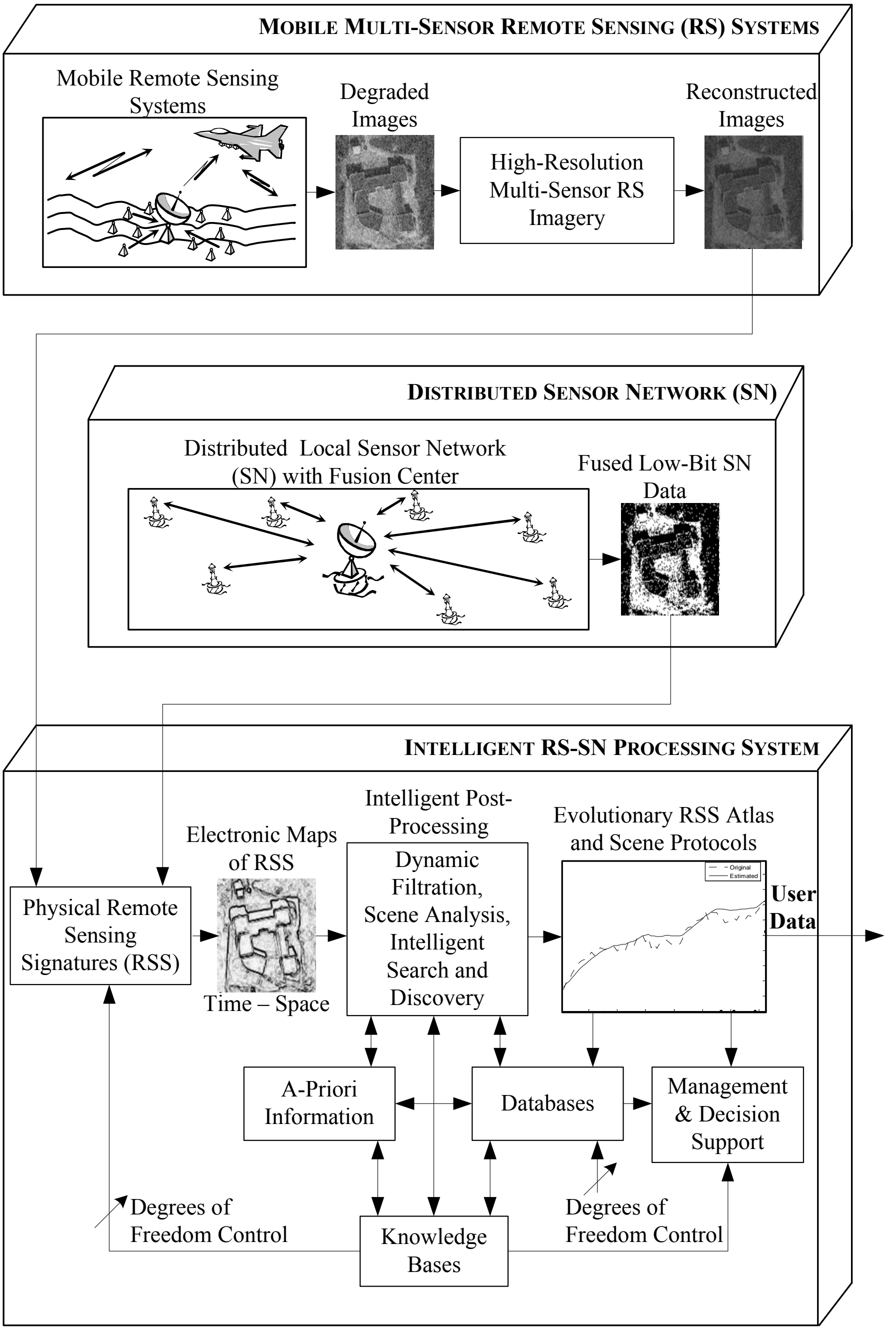

MOBILE MULTI-SENSOR REMOTE SENSING (RS) SYSTEMS
Mobile Remote Sensing Systems
Degraded Images
High-Resolution Multi-Sensor RS Imagery
Reconstructed Images
DISTRIBUTED SENSOR NETWORK (SN)
Distributed Local Sensor Network (SN) with Fusion Center
Fused Low-Bit SN Data
INTELLIGENT RS-SN PROCESSING SYSTEM
Physical Remote Sensing Signatures (RSS)
Electronic Maps of RSS
Time – Space
Intelligent Post-Processing
Dynamic Filtration, Scene Analysis, Intelligent Search and Discovery
Evolutionary RSS Atlas and Scene Protocols
Original
Estimated
User Data
A-Priori Information
Databases
Management & Decision Support
Knowledge Bases
Degrees of Freedom Control
Degrees of Freedom Control

Fig. 1. Block-diagram of the proposed intelligent DEDR approach with KB multisensor data fusion

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} E(\mathbf{v} \mid \boldsymbol{\lambda}) \tag{1}$$

of minimization of the aggregated objective (cost) function

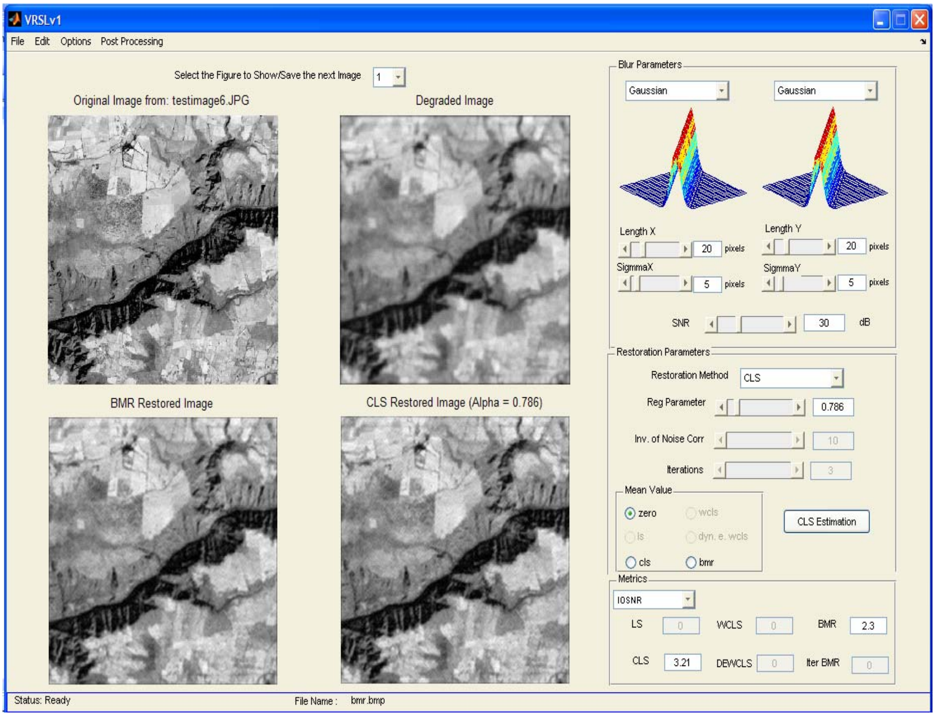
$$E(\mathbf{v} \mid \boldsymbol{\lambda}) = -H(\mathbf{v}) + (1/2)\sum_{m=1}^{M} \lambda_m J_m(\mathbf{v}) + (1/2)\lambda_{M+1} J_{M+1}(\mathbf{v}) \tag{2}$$

with respect to the desired K-D image vector $\mathbf{v}$ for the assigned (or adjusted) values of M+1 regularization parameters $\{\lambda_m\}$ that compose a vector of the controllable algorithmic "degrees of freedom" $\boldsymbol{\lambda}$. In a particular employed method, the proper selection of $\{\lambda_m\}$ is associated with the parametric-level adjustment of the SP optimization procedure (2). Here, $H(\mathbf{v}) = -\sum_{k=1}^{K} v_k \ln v_k$ represents the image entropy [3], $\{J_m(\mathbf{v})\}$ (m = 1, …, M) compose a set of particular objective (cost) functions incorporated into the optimization, and $J_{M+1}(\mathbf{v})$ represents the additional regularizing stabilizer [3] that controls specific metrics properties of the desired image. The data acquisition model is defined by the set of equations, $\mathbf{u}^{(m)} = \mathbf{F}^{(m)}\mathbf{v} + \mathbf{n}^{(m)}$ for M methods/systems to be aggregated/fused, i.e. m = 1, …, M, where $\mathbf{F}^{(m)}$ represent the system/method degradation operators usually referred to as the imaging system point spread functions (PSF), and vectors $\mathbf{n}^{(m)}$ represent composite noises (usually with unknown statistics) in the actually acquired images, respectively.

Different RS imaging methods incorporate different definitions for corresponding employed objective (cost) functions $\{J_m(\mathbf{v})\}$ [2], [3], [5], [6], [8], [10]. For the deterministic constrained least squares (CLS) method [2], [3], $J_m(\mathbf{v}) = ||\mathbf{u}^{(m)} - \mathbf{F}^{(m)}\mathbf{v}||^2$, are associated with partial error functions. For the weighted CLS (WCLS) method [6], the objective costs incorporate the user-defined weight matrices $\{\mathbf{W}_m\}$ as additional "degrees of freedom", i.e. $J_m(\mathbf{v}) = ||\mathbf{u}^{(m)} - \mathbf{F}^{(m)}\mathbf{v}||^2{}_{\mathbf{W}m}$. The unified DEDR paradigm incorporates into the unified optimization problem (1), (2) also other robust and more sophisticated statistical methods, among them are: the rough conventional matched spatial filtering (MSF) approach [8]; the descriptive maximum entropy (ME) technique [6]; the robust spatial filtering (RSF) method [5], the robust adaptive spatial filtering (RASF) technique [8], the fused Bayesian-DEDR regularization (FBR) method [5], the POCS-regularized DEDR method, i.e., the unified DEDR-POCS [10]; etc. All such methods involve particular specifications of the corresponding $\{J_m(\mathbf{v})\}$ and $\{\mathbf{W}_m\}$ into the DEDR optimization procedure (1), (2). It is important to note that due to the non-linearity of the objective function (2) the solution of the parametrically controlled fusion-optimization problem (1), (2) will require extremely complex (NP-complex [10]) algorithms and result in the technically intractable computational schemes if solve these problems employing the standard direct minimization techniques [1], [3]. For this reason, we propose to apply the POCS-regularized fixed-point iterative techniques implemented using the neural network (NN) based computing for solving such aggregated DEDR optimization problems (1), (2) as those were detailed in the previous studies [3], [10], [11].

4 Integrated VRSL Software

Having developed a manifold of the DEDR-POCS computational techniques, the next goal is to computationally implement, verify, and demonstrate the capabilities of the collaborative RS signal and image processing for RSS extraction, KB intelligent scene analysis, multiple target detection and scene zones localization via development of the sophisticated end-user-oriented software that we refer to as "Virtual remote sensing laboratory" (VRSL). The purpose of the elaborated VRSL software is to implement computationally all considered DEDR-related methods (MSF, CLS, WCLS, ME, RSF, RASF, FBR, etc) and to perform the RS image formation/ reconstruction/enhancement tasks with or without method and/or sensor system fusion. The VRSL software (created in the MATLAB V.7 computational environment) aggregates interactive computational tools that offer to the user different options of acquisition and processing of any image in the JPEG, TIFF, BMP and PNG formats as test input images, application of different system-level effects of image degradation with a particular simulated RS system, simulation of random noising effects with different noise intensities and distributions. Next, various RS image enhancement/fusion/reconstruction/post-processing tasks can be simulated in an interactive mode applying different DEDR-related algorithms to the degraded noised images, and the quantitative performance enhancement characteristics attained in every particular simulated scenario can then be computed and archived [11].



Select the Figure to Show/Save the next Image: 1

Blur Parameters: Gaussian, Gaussian

Length X: 20 pixels
Length Y: 20 pixels
SigmmaX: 5 pixels
SigmmaY: 5 pixels
SNR: 30 dB

Restoration Method: CLS
Reg Parameter: 0.786
Inv. of Noise Corr: 10
Iterations: 3

Mean Value: ● zero ○ wcls ○ ls ○ dyn. e. wcls ○ cls ○ bmr

Metrics: IOSNR
LS: 0
WCLS: 0
BMR: 2.3
CLS: 3.21
DEWCLS: 0
Iter BMR: 0

Status: Ready
File Name : bmr.bmp

Fig. 2. Elaborated graphical user interface of the VRSL

The user has options to display on the screen all simulated processed scene images and RSS along with the corresponding protocols of analysis of different performance quality metrics (see the illustrative RS image reconstruction examples displayed in the user interface presented in Fig. 2).

5 Simulation Examples

Here, we report some simulations of the DEDR-related algorithms (performed using the elaborated VRSL) carried out in two dimensions for the uncertain operational scenario with the randomly perturbed SFO, in which case the measurement data are contaminated by the composite speckle and multiplicative signal-dependent noise [9], [10]. The simulation experiments that we report in this paper relate to enhancement of the RS images acquired with a fractional SAR system characterized by the PSF of a Gaussian "bell" shape in both directions of the 2-D scene (in particular, of 16 pixel width at 0.5 from its maximum for the 512-by-512 pixel-formatted test scenes). The chi-squared additive noise with the SNR = 15 dB was incorporated to test and compare the performances of different employed enhancement methods. Two scenes (the artificially synthesized and borrowed from the real-world RS imagery) were tested. These are displayed in Figures 3(a) and 3(b), respectively. The qualitative simulation results for six different DEDR-related enhancement/reconstruction procedures for the first test scene are presented in Fig. 4 and for the second scene in Fig. 5, respectively, with the corresponding IOSNR (improvement in the output signal-to-noise ratio [8]) quantitative performance enhancement metrics reported in the figure captions (in the [dB] scale).

From the reported simulation results, the advantage of the well designed imaging experiments (POCS-regularized DEDR, ASF and adaptive RASF) over the case of badly designed experiment (non-robust MSF, de-speckling without DEDR enhancement and non-constrained RSF) is evident for both scenes. Due to the performed regularized inversions, the resolution was substantially improved in both tested scenarios. The higher values of *IOSNR* were obtained with the adaptive robust DEDR-related estimators, i.e. with the POCS-regularized iterative fixed-point DEDR technique empirically adapted to the uncertain operational scenario.

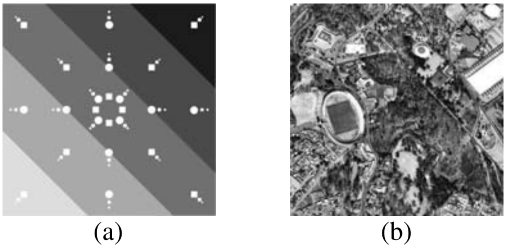


Fig. 3. Original test scenes: (a) artificially synthesyzed scene; (b) real-world RS scene borrowed from the high-resolution RS imagery [12]. These test scenes are not observable with the simulated SAR imaging system that employs the conventional MSF image formation method.

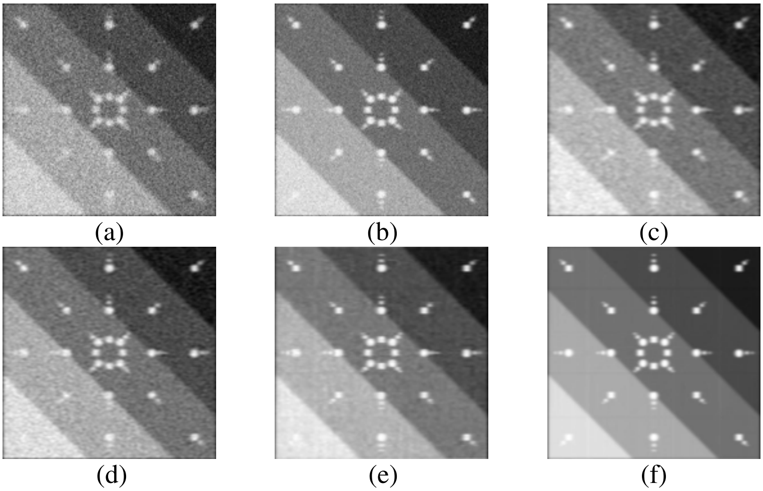


Fig. 4. Simulation results for the artificially synthesized scene: (a) degraded SAR scene image formed applying the MSF method [*IOSNR* = 0 dB]; (b) adaptively de-speckled MSF image [*IOSNR* = 0.62 dB]; (c) image reconstructed applying the non-constrained RSF algorithm [*IOSNR* = 7.24 dB]; (d) image reconstructed with the constrained RSF algorithm [*IOSNR* = 8.35 dB]; (e) image reconstructed applying the non-constrained ASF algorithm [*IOSNR* = 9.41 dB]; (f) image reconstructed applying the POCS-regularized adaptive DEDR method [*IOSNR* = 15.70 dB]

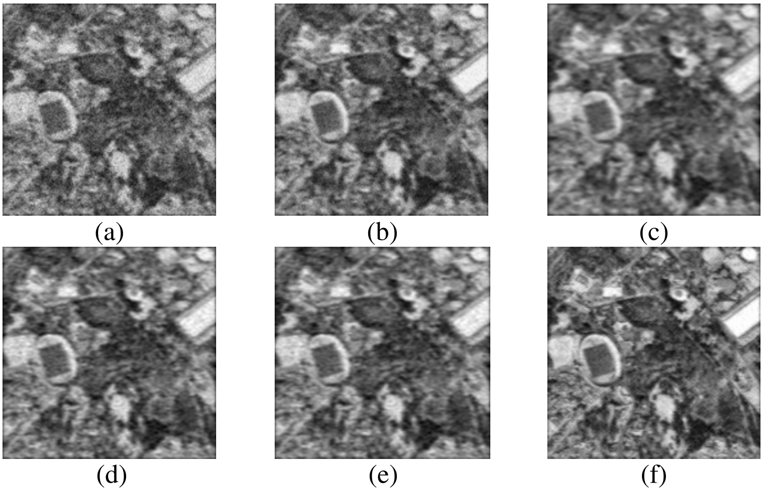


Fig. 5. Simulation results for the real-world RS test scene: (a) degraded SAR scene image formed applying the MSF method [*IOSNR* = 0 dB]; (b) adaptively de-speckled MSF image [*IOSNR* = 0.62 dB]; (c) image reconstructed applying the non-constrained RSF algorithm [*IOSNR* = 6.33 dB]; (d) image reconstructed with the constrained RSF algorithm [*IOSNR* = 7.78 dB]; (e) image reconstructed applying the non-constrained ASF algorithm [*IOSNR* = 9.14 dB]; (f) image reconstructed applying the POCS-regularized adaptive DEDR method [*IOSNR* = 14.33 dB]

6 Concluding Remarks

The descriptive experiment design regularization (DEDR) approach for high-resolution estimation of spatial RS signature fields has been unified with the KB post-processing paradigm for the purposes of high-resolution RS imaging, search, discovery, discrimination, mapping and problem-oriented analysis of the diverse RS data. To accomplish computationally different DEDR-specified numerical optimization and processing tasks we have elaborated and reported the end-user-oriented VRSL software. The VRSL provides the necessary tools for computer-aided simulation and analysis of different DEDR-related RS image formation/enhancement/ reconstruction/fusion/post-processing techniques developed using the unified KB DEDR methodology. The reported simulation results are illustrative of the VRSL usefulness and capabilities in computer simulations of different RS imaging tasks performed with the artificial and real-world RS imagery.

References

1. Falkovich, S.E., Ponomaryov, V.I., Shkvarko, Y.V.: Optimal Reception of Space-Time Signals in Channels With Scattering (in Russian). Radio i Sviaz, Moscow (1989)
2. Wehner, D.R.: High-Resolution Radar, 2nd edn. Artech House, Boston (1994)
3. Barrett, H.H., Myers, K.J.: Foundations of Image Science. Willey, New York (2004)
4. Ishimary, A.: Wave Propagation and Scattering in Random Media. IEEE Press, NY (1997)
5. Shkvarko, Y.: Estimation of wavefield power distribution in the remotely sensed environment: Bayesian maximum entropy approach. IEEE Trans. Signal Proc. 50(9), 2333–2346 (2002)
6. Shkvarko, Y.: Unifying regularization and Bayesian estimation methods for enhanced imaging with remotely sensed data—Part I: Theory. IEEE Trans. Geoscience and Remote Sensing 42(5), 923–931 (2004)
7. Shkvarko, Y.: Unifying regularization and Bayesian estimation methods for enhanced imaging with remotely sensed data—Part II: Implementation and performance issues. IEEE Trans. Geoscience and Remote Sensing 42(5), 932–940 (2004)
8. Shkvarko, Y.: From matched spatial filtering towards the fused statistical descriptive regularization method for enhanced radar imaging. EURASIP J. Applied Signal Processing 2006, 1–9 (2006) Article ID 39657
9. Greco, M.S., Gini, F.: Statistical analysis of high-resolution SAR ground clutter data. IEEE Trans. Geoscience and Remote Sensing 45(3), 566–575 (2007)
10. Shkvarko, Y., Perez-Meana, H., Castillo-Atoche, A.: Enhanced radar imaging in uncertain environment: A descriptive experiment design regularization approach. Int. J. Navigation and Observation 2008, 1–11 (2008) Article ID 810816
11. Shkvarko, Y., Gutierrez-Rosas, J.A., de Guerrero Diaz Leon, L.G.: Towards the virtual remote sensing laboratory: Simulation software for intelligent post-processing of large scale remote sensing imagery. In: Proc. IEEE Intern. Symposium on Geoscience and Remote Sensing, IGARSS 2007, Barcelona, Spain, pp. 1561–1564 (2007)
12. Space Imaging, GeoEye Inc. (2009), http://www.spaceimaging.com/quicklook

XIX CASI 2009 Workshop II: Intelligent Fussion and Classification Techniques

Optimizing Classification Accuracy of Remotely Sensed Imagery with DT-CWT Fused Images

Diego Renza[1], Estibaliz Martinez[2], and Agueda Arquero[2]

[1] National University of Colombia
drenzat@unal.edu.co
[2] Polytechnic University of Madrid
{emartinez,aarquero}@fi.upm.es
http://www.fi.upm.es

Abstract. Image fusion is a basic tool for combining low spatial resolution multi-spectral and high spatial resolution panchromatic images using advanced image processing techniques. Study on efficient image fusion method for specific application is one of the most important objectives in current remote sensing community. On the other hand, it is well known that the image classification techniques combine complex processes that may be affected by factors like the resolution of remote sensed images. This study focuses on the influence of image fusion on spectral classification algorithms and their accuracy. Results are presented on SPOT images. The best results were achieved by Dual Tree Complex Wavelet Transform (DT-CWT)).

Keywords: Classification accuracy, Image fusion, Dual Tree Complex Wavelet Transform, DT-CWT.

1 Introduction

Remote sensing is a fundamental tool providing a relatively lower cost for the detection, classification and monitoring of landslide phenomena.Classification aims to convert the remotely sensed image into a thematic map that depicts the spatial distribution of the various land-cover classes found within the region. These maps have been used to analyze the impacts of land use change on the environment, improve land use planning and natural resource management, and better understand ecological processes on Earth. Of the many classification approaches available, most researchers use either supervised or unsupervised classification. Supervised classification consists of two stages: training and classification. In the training stage, the analyst identifies representative training areas and develops a numerical description of the spectral attributes of each land cover type of interest in the scene. In the classification stage, each pixel in the image data set is categorized into the land cover class it most closely resembles [1]. Unsupervised classification techniques do not utilize training data as the basis for classification. Usually supervised classifiers are preferred to unsupervised clustering algorithms, which are intrinsically less suitable to obtain accurate classification maps.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1031–1038, 2009.

Classification of remote sensing images is a complex task whose accuracy strongly depends on the available prior information. In this sense, given the general high complexity of the problem, we need to dispose of high quality images. Image fusion techniques are useful to integrate the geometric detail of high spatial resolution panchromatic (PAN) image with the colour information of high spectral resolution multispectral (MS) image to produce a MS image with high spatial quality. High spatial resolution MS image data are indispensable for inventorying and monitoring of land surface phenomena [2].

A recent study shows the influence of image fusion on spectral classification for Landsat images, with acceptable results [3]. This paper analyzes the accuracy results of image supervised classification, which can be improved by using fused SPOT images obtained through multi-resolution analysis (MRA) by Dual Tree Complex Wavelet Transform (DT-CWT), compared with classical fusion algorithms.

2 Background

2.1 Fusion Methods

The advantage of a fusion process is that a single image can be achieved containing both the high spatial resolution and spectral information, hence, the result of image fusion is a new image which is more suitable for human and machine perception or further image-processing tasks such a classification, segmentation, feature extraction or object recognition. There are a variety of types of fusion, these range from a simple average of the pixels in registered images to a more complex schemes using multiresolution analysis (MRA) (pyramids, wavelets, etc), this variety of schemes have been developed over the last 25 years, with classical approaches such as IHS (Hue-Intensity-Saturation), PCA (Principal Component analysis) or HPF (High pass filter).

One step beyond, are the multi-resolution analysis (MRA), which provides effective tools to facilitate the implementation of data fusion and have shown a better performance when injecting the high frequency information extracted from de PAN into resampled versions of MS [4], by example A Trous Wavelet transform [1], Curvelet Transform [2], Hermite transform [3]. Here, it is necessary to consider that when injecting high-pass details, spatial distortions may occur, resulting in translations or blurring of the contours and textures, this occurs mainly when the MRA is not shift invariant [4].

There is a MRA algorithm, that is particularly suitable for images and other multi-dimensional signals processing, which is approximate shift invariance and computational efficiency with good well-balanced frequency responses, this will be described below.

[1] Zhu et al: Fusion of High-Resolution Remote Sensing Images Based on a trous Wavelet Algorithm, 2004.

[2] Nencini et al: Remote sensing image fusion using the curvelet transform, 2007.

[3] Escalante et al, A.: The Hermite Transform An Efficient Tool for Noise Reduction and Image Fusion in Remote-Sensing, 2006).

2.2 Dual Tree Complex Wavelet Transform

An effective approach for conducting an analytical wavelet transform, originally introduced by Kingsbury in 1998, is called Dual Tree Complex Wavelet Transform, avoid unwanted frequency component and is also expansive, but only by a factor 2 (1-D signals), independent of the number of stages. The properties that characterize the DT-CWT are highlighted in several papers [5]. The principal properties are: good shift invariance, good directional selectivity in m-D, perfect reconstruction with short support filters, limited redundancy (2:1 in 1-D, 4:1 in 2-D, etc) and low computation, much less than the undecimated (a trous) DWT.

The Framework of the Dual-Tree. The DT-CWT uses two real DWTs, the first DWT delivers real part of transform while the second DWT delivers the imaginary part. FBS for the analysis and synthesis used to implement the dual-tree CWT and its inverse are shown in Figure 1.

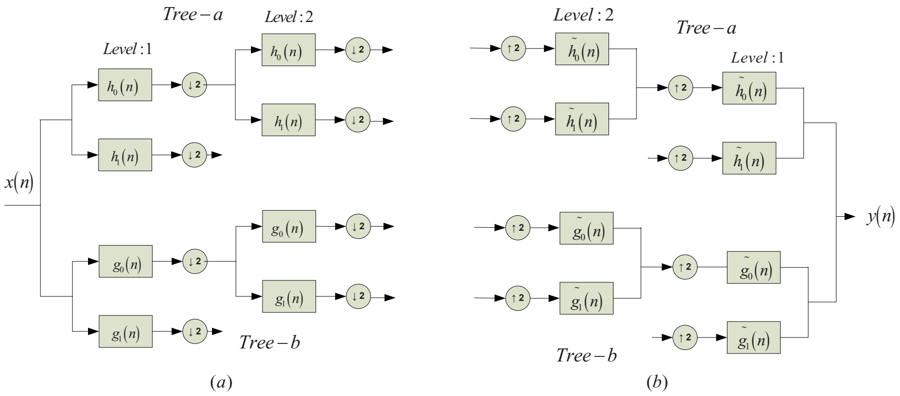


Fig. 1. Filter Bank (FB) for DT CWT (a) Analysis (b) Synthesis

Here, $h_0(n)$, $h_1(n)$ are the pair of low pass filter / high pass to the superior FB (tree a), $g_0(n)$, $g_1(n)$ the pair of low pass filter / high pass for inferior FB (tree b). $\psi_h(t)$ y $\psi_g(t)$ are the two real wavelets associated with each of the two transform (tree a and b, respectively), below there are the features for the DT-CWT's filter banks.

- The two real wavelet transforms use two different sets of filters, each of whom meets the conditions for PR.
- In addition to satisfying the PR conditions, the filters are designed so that the complex wavelet $\psi := \psi_h(t) + j\psi_h(t)$ is approximately analytic.
- According to the above, are designed so that $\psi_g(t)$ is approximately the Hilbert transform of $\psi_h(t)$ (denoted by $\mathrm{H}\{\psi_h(t)\}$).

- To satisfy this condition, one of the filters must have a displacement approximately half-sample with respect to the other [5]. Recent studies show alternatives to this classical approach[4],[5],[6].
- The filters themselves are real.
- The transform is twice expansive (not critically sampled).

3 Methodology

3.1 Scene Study

A multispectral (MS) and panchromatic (PAN) SPOT images were used in this study. The images have a spatial resolution of 10 m and 2.5 m and covered an area of the order of 26 km^2. The scene consists of three great environmental blocks: to the left there are extensions of natural mediterranean forest partially degraded, in the central axis and crossing all the scene it is characterized by the city-planning, industrial advance and road infrastructures that degrade zones of agrarian development in other times and the right part are the vestiges of agricultural zones as a result of the Jarama river presence, one important of the community of Madrid (Spain). The left corner is placed at 450775 E and 4494685 N (UTM geographic coordinates, h30).

Eleven thematic classes have been supervised in situ: the water class corresponding to a reservoir and in a golf course; five vegetation types like the natural of degraded Mediterranean forest (oak forest), two zones of irrigated land crop, green of golf course and riverside vegetation; mixed area with vegetation and ground; clearly built-up ground with highway and three types of ground as soil in fallow land and without specific use. Thus, the names asigned for training areas have been: water, natural soil, crop1, crop2, green-golf, riverside vegetation, mixed soil, urban soil, soil1, soil2 and soil3.

3.2 Fusion Process and Quality Determination

From the increasing variety of image fusion methods, a selection oh three fusion techniques was chosen, Principal Components Analysis (PCA) and two based on multi-resolution analysis: A Trous Wavelet Transform (AWT3) (3 levels) and Dual Tree complex Wavelet Transform (3 levels). For first two, we used the IJFusion framework[7]; for the last one, we propose a method using the toolbox for DT-CWT, provided by the Dr. Nick Kingsbury[8]. This method will be described below.

[4] Tayet al.: Orthonormal Hilbert-Pair of Wavelets With (Almost) Maximum Vanishing Moments, 2006.

[5] Bogdan et al.: Optimization of Symmetric Self-Hilbertian Filters for the Dual-Tree Complex Wavelet Transform, 2008.

[6] Yu et al.: Sampled-Data Design of FIR Dual Filter Banks for Dual-Tree Complex Wavelet Transforms via LMI Optimization, 2008.

[7] http://www.ijfusion.es/

[8] ngk@eng.cam.ac.uk

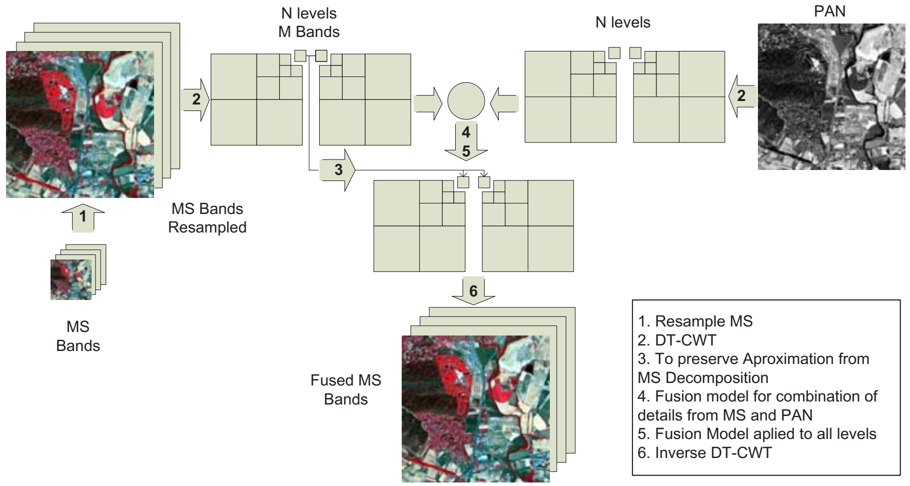

N levels
M Bands
N levels
PAN
2
2
1
3
4
5
6
MS Bands
Resampled
MS
Bands
Fused MS
Bands
1. Resample MS
2. DT-CWT
3. To preserve Aproximation from MS Decomposition
4. Fusion model for combination of details from MS and PAN
5. Fusion Model aplied to all levels
6. Inverse DT-CWT

Fig. 2. Image Fusion with DT-CWT

Image Fusion with DT-CWT. The fusion process involves the following steps, see Figure 2.

1. The low-resolution MS image is resample to the size of the panchromatic image, after this process, the 2 images are the same size.
2. Implement the dual tree wavelet transform to MS bands and PAN image, a certain number of levels, for this particular case, we use 3 levels of decomposition.
3. Defines the coefficients of the DT-CWT decomposition for a new high-resolution MS image (HRMS). For these new coefficients is preserved approximation of the MS decomposition.
4. To calculate the details of the new HRMS are considered both the wavelet coefficients of the PAN as the MS, i.e., injecting the details of the PAN according to a weight that depends on the ratio of standard deviations for a neighborhood around the coefficient in question.
5. This procedure is carried out recursively from the lowest to the highest level.
6. Finally apply the inverse DT-CWT to new (HRMS) coefficients.
7. New bands come together to compose the merged image, this image not only contains the original spectral information but also the structure of information in the panchromatic image, i.e., improving both the spatial and spectral information of original images.

Image Fusion Assessment. To know the quality of the merged image, the new fused image is usually compared against a reference image (true), to create it, and having the sets of original images A_h and B_{KL}, where A denotes the PAN image, B the MS image, h high resolution, l low resolution and K each band of the MS; the image A_h is degraded to the low resolution l, giving A_l. B_{KL} images

Table 1. Metrics used for fusion assessment

Correlation Coefficient (CC) [6]	$CC\left(X,Y,k\right)=\frac{\sum\limits_{m}\sum\limits_{n}\left[\left(X_{mn}\left(k\right)-X'_{mn}\right)\left(Y_{mn}\left(k\right)-Y'_{mn}\right)\right]}{\sqrt{\sum\limits_{m,n}\left(X_{mn}\left(k\right)-X'_{mn}\right)^{2}\sum\limits_{m,n}\left(Y_{mn}\left(k\right)-Y'_{mn}\right)^{2}}}$ (1)
	K: k-band, X: Original image, Y: Merged image, MxN: Image dimensions
ERGAS [7]	$ERGAS=100\frac{d_h}{d_l}\sqrt{\frac{1}{L}\sum\limits_{l=1}^{L}\left(\frac{RMSE\left(l\right)}{\mu\left(l\right)}\right)^{2}}$ (2)
	d_h :PAN pixel size (m), d_l: MS pixel size (m) L**:** Number of spectral bands, $\mu\left(l\right)$.Mean of each spectral band, $RMSE\left(l\right)$**:** *Root Mean Squared Error*
Spatial CC (SCC)	$CC\left(X_hf,Y_hf,k\right)$ (3)
	X_{hf}, Yhf: High frequency image, obtained by means a Laplacian filter

are degraded to the resolution $l(l/h)$, giving B'_{kl}. The fusion process is applied to images A_l and B'_{KL}, the quality can be assessed using as a reference image the original MS image, i.e. B_{kl}. The metrics used are listed in the Table 1.

3.3 Classification Process

We employed a consistent classification scheme applied to each fusion result using two supervised parametric and non-parametric classifiers. Within the parametric classification approaches, the maximum likelihood classification (MLC) algorithm [8] is one of the most applied methods. In contrast to the maximum likelihood classifier, support vector machines (SVM) [9] can be used as nonparametric or discriminative, binary classifiers. In this study, the two classifiers above mentioned have been used.

Since supervised methods have been considered, the first step is the definition of a set of training areas in the images to be classified. This process is critical because these areas would be used like an estimator of the land cover classes. The selection of training areas has been carried out through a two-dimensional scatter diagram (scattergram) methodology [10].

Then for supervised classification, the training areas were selected according to eleven thematic classes derived by scattergram methodology, visual image inspection and field work. The total number of training areas pixels were 130 per class approximately. Additionally, a 10% of test samples for accuracy assessment were collected and treated separately. For fused images, the training and test areas were selected in the same geographical location. Each classification was performed and assessed with identical training and test samples.

When the remote sensing images were classified, the accuracy of each classification was assessed using statistical method [11]. Error matrices were constructed, then overall accuracy (OA) and the kappa index (K) were calculated to assess the whole accuracy of the classified map, and the producer accuracy and user accuracy were used to interpret the success of each class.

4 Results

The Table 2 shows the results in each band (b1:green, b2:red, b3:NIR y b4:SWIR) for the quality indexes used in this work. The proposed method (DT-CWT3) shows high spectral quality with high spatial quality; better results in spatial quality were achieved with the simple injection of PAN details (DT-CWT3P), but with a slightly lower spectral quality. In general terms, the DT-CWT is a good approach for image fusion, reaching better results than traditional schemes.

Table 2. Results for image fusion

	ERGAS	**CC**				**SCC**			
	b1-b4	**b1**	**b2**	**b3**	**b4**	**b1**	**b2**	**b3**	**b4**
PCA	7.4184	0.9701	0.9739	0.9196	0.938	0.9944	0.9982	0.9575	0.9924
AWT3	1.7987	0.9737	0.9863	0.7893	0.8776	0.9995	0.9994	0.9993	0.9994
DT-CWT3P	1.7581	0.9735	0.9884	0.8104	0.883	0.9994	0.9993	0.9982	0.9989
DT-CWT3	1.3945	0.9827	0.987	0.8653	0.9216	0.9946	0.994	0.9928	0.9941

The results of the accuracy classification are showed in the Table 3. In all cases, the overall accuracy (OA) and the kappa (K) index are good. The best results are achieved in the classifications of fused images obtained by DT-CWT. In general, user's accuracy of natural soil class is lower (60.87-82.35%) than the others classes. There is a considerable confusion with riverside vegetation class in the case of AWT3 fusion. This mis-assignment was caused by the mixed and variable spectral response of the natural soil class, which comprises a wide variety of surface cover types.

Table 3. Accuracy Parameters of Image Classification

	MULTI		PCA-Fus		AWT3-Fus		DT-CWT3-Fus	
	MLC	SVM	MLC	SVM	MLC	SVM	MLC	SVM
OA (%)	95.6	97.2	94.2	96.7	95.6	95.4	98.5	97.5
K	0.951	0.969	0.936	0.964	0.951	0.949	0.984	0.973

5 Conclusions

In this study focused on the influence of image fusion approaches on classification accuracy, three fusion approaches were applied to a SPOT image. The fusion results were employed for supervised classification with two methods (MLC and

SVM). High values of classification accuracies of fused and no fused images are obtained. From evaluations with fused images, the following ranking was derived (from best to poorest): Dual Tree Complex Wavelet Transform (DT-CWT), A Trous Wavelet Transform (AWT) and Principal Component Analysis (PCA).

Acknowledgments. The authors would like to thank Dr. Nick Kingsbury (from the Signal Processing Group, Dept. of Eng., University of Cambridge) for the DT-CWT source code, and to Dr. Boris Escalante (from National Autonomous University of Mexico) for advice in the process fusion.

References

1. Lillesand, T.M., Kiefer, R.W., Chipman, J.W.: Remote Sensing and Image Interpretation, 5th edn., p. 763. Wiley, New York (2004)
2. Oguru, Y., Takeuchi, S., Ogawa, M.: Adv. Space Res. Higher Resolution Images for visible and near infrared bands of Landsat-7 ETM+ by using Panchromatic Band 32(11), 2269–2274 (2003)
3. Colditz, R., Wehrmann, T., Bachmann, M., Steinnocher, K., Schmidt, M., Strunz, G., Dech, S.: Influence of image fusion approaches on classification accuracy: a case study. Int. Journal of Remote Sensing 27(15), 3311–3335 (2006)
4. Alparone, L., Wald, L., Chanussot, J., Thomas, C., Gamba, P., Bruce, L.M.: Comparison of Pansharpening Algorithms Outcome of the 2006 GRS-S Data-Fusion Contest. IEEE Tran. on Geoscience and Remote Sensing 45(10) (2007)
5. Selesnick, I.W., Baraniuk, R.G., Kingsbury, N.G.: The Dual-Tree Complex Wavelet Transform. IEEE Signal processing magazine, 123–151 (2005)
6. Jian, M., Dong, J., Zhang, Y.: Image fusion based on wavelet transform. In: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (2007)
7. Wald, L.: Assessing the quality of synthesized images, Chapter 8 of Data Fusion. Definitions and Architectures - Fusion of Images of Different Spatial Resolutions. Presses de l'Ecole, Ecole des Mines de Paris, Paris, France (2002)
8. Richards, J.A.: Remote sensing digital image analysis, p. 340. Springer, Berlin (1995)
9. Vapnik, V.N.: The Nature of statistical learning theory, p. 214. Springer, Berlin (1995)
10. Martinez, E., Gonzalo, C., Arquero, A., Gordo, O.: Evaluation of different Fuzzy knowledge acquisition methods for remote sensing image classification. In: Remote Sensing of the System Earth- A Challenge for the 21th Century 1999 IGARSS 1999, vol. 5, pp. 2489–2491 (1999)
11. Liu, C., Frazier, P., Kumar, L.: Comparative assessment of the measures of thematic classification accuracy. Rem. Sens. of Environment 107, 606–616 (2007)

Filter Banks for Hyperspectral Pixel Classification of Satellite Images

Olga Rajadell, Pedro García-Sevilla, and Filiberto Pla

Depto. Lenguajes y Sistemas Informáticos
Jaume I University, Campus Riu Sec s/n 12071 Castellón, Spain
{orajadel,pgarcia,pla}@lsi.uji.es
http://www.vision.uji.es

Abstract. Satellite hyperspectral imaging deals with heterogenous images containing different texture areas. Filter banks are frequently used to characterize textures in the image performing pixel classification. This filters are designed using different scales and orientations in order to cover all areas in the frequential domain. This work is aimed at studying the influence of the different scales used in the analysis, comparing texture analysis theory with hyperspectral imaging necessities. To pursue this, Gabor filters over complex planes and opponent features are taken into account and also compared in the feature extraction process.

1 Introduction

Nowadays imaging spectrometers are significantly increasing their spatial resolution. As their resolution increases, smaller areas are represented by each pixel in the images, encouraging the study of the relations of adjacent pixels (texture analysis) [9] [6]. However, not only the spatial resolution increases but also the spectral resolution. This entails dealing with a large number of spectral bands with highly correlated data [7].

Both dimensionality and texture analysis in hyperspectral imaginary have been tackled from different points of view in literature. Several solutions to the dimensionality problem can be found, such as selection methods based on mathematical dimensionality reduction [10] or methods based on information theory which try to maximize the information provided by different sets of spectral bands [7].

Moving to texture analysis, literature survey provides us with a wide variety of well known texture analysis methods based on filtering [8] [4]. It is well known that, when dealing with microtextures, the most discriminant information falls in medium and high frequencies [1] [9]. It has been recently proposed that spatial/texture analysis may significantly improve the results in pixel classification tasks for satellite images using a very reduced number of spectral bands [11]. Therefore, it may be convenient to identify the influence of each frequency band separately in order to compare the textural information with the specific necessities of hyperspectral satellite imaging.

Besides, color opponent features were first introduced in color texture characterization with fairly good performance [3] and later extended to deal with multi-band texture images [4]. However, they have never been used to perform pixel classification tasks in satellite images. In this paper, we study several Gabor filter banks as well as multi-band opponent features for pixel classification tasks.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1039–1046, 2009.

2 Filter Banks and Feature Extraction

Applying a filter over an image band provides a response for each pixel. If a filter bank is applied, a pixel can be characterized by means of the responses generated by all filters. It is possible to apply a filter in the space domain by a convolution or in the frequency domain by a product. In both cases, the response is the corresponding part of the original pixel value which responds to the filter applied [12].

When using filter banks, they are generally designed considering a dyadic tessellation of the frequency domain, that is, each frequency band (scale) considered is double the size of the previous one. It should not be ignored that this tessellation of the frequency domain thoroughly analyzes low frequencies giving less importance to medium and higher frequencies. Because the purpose of this work is to study the importance of texture in the pixel classification task, an alternative constant tessellation (given the same width to all frequency bands) is proposed in order to ensure an equal analysis of all frequencies.

2.1 Gabor Filters

Gabor filters consist essentially of sine and cosine functions modulated by a Gaussian envelope that achieve optimal joint localization in space and frequency. They can be defined by eq. (1) and (2) where m is the index for the scale, n for the orientation and u_m is the central frequency of the scale.

$$f_{mn}^{real}(x,y) = \frac{1}{2\pi\sigma_m^2} \exp\left\{-\frac{x^2+y^2}{2\sigma_m^2}\right\} \times \cos(2\pi(u_m x \cos\theta_n + u_m y \sin\theta_n)) \quad (1)$$

$$f_{mn}^{imag}(x,y) = \frac{1}{2\pi\sigma_m^2} \exp\left\{-\frac{x^2+y^2}{2\sigma_m^2}\right\} \times \sin(2\pi(u_m x \cos\theta_n + u_m y \sin\theta_n)) \quad (2)$$

If symmetrical filters are considered only the real part must be taken into account.

2.2 Gabor Filters over Complex Planes

Texture analysis in multi-channel images has been generally faced as a multi-dimensional extension of techniques designed for mono-channel images. In this way, images are decomposed into separated channels and the same feature extraction process is performed over each channel. This fails in capturing the interchannel properties of a multi-channel image.

To describe the inter-channel properties of textures we propose features obtained using Gabor filters over complex planes. This means that instead of using each spectral band individually, we take advantage of the complex definition and introduce the data of two spectral bands into one complex band, one as the real part and the other one as the imaginary part. In this way we involve pairs of bands in each characterization process, as it happens for the opponent features. As a result, for a cluster of spectral bands, we will consider all possible complex bands (pairs of bands). The Gabor filter bank will be applied over all complex bands as shown in eq. 3, where $I^i(x,y)$ is the i^{th} spectral band.

$$h_{mn}^{ij}(x,y) = (I^i(x,y) + I^j(x,y)i) * f_{mn}(x,y) \quad (3)$$

The feature vector for each pixel in the image is composed of the response for that pixel to all filters in the filter bank, that is:

$$\psi_{x,y} = \{h^{ij}_{mn}(x,y)\}_{\forall i,j/i\neq j,\forall m,n} \tag{4}$$

The size of the feature vector varies with the number of complex bands. For each complex band, one feature is obtained for each filter applied what means that there will be as many features as filters for each complex band and as many complex bands as combinations without order nor repetition may be done with two bands in the cluster B. The total number of features is given by eq. 5 where M stands for the number of scales and N for the number of orientations.

$$size(\psi_{x,y}) = M \times N \times \binom{B}{2} \tag{5}$$

2.3 Opponent Features

Opponent features combine spatial information across spectral bands at different scales and are related to processes in human vision [3]. They are computed from Gabor filters as the difference of outputs of two different filters. The combination among filters are made for all pair of spectral bands i, j with $i \neq j$ and $|m - m'| \leq 1$:

$$d^{ij}_{mm'n}(x,y) = h^{i}_{mn}(x,y) - h^{j}_{m'n}(x,y) \tag{6}$$

In this case, the feature vector for a pixel is the set of all opponent features for all spectral bands.

$$\varphi_{x,y} = \{d^{ij}_{mm'n}(x,y)\}_{\forall i,j/i\neq j,\forall m,m'/|m-m'|\leq 1,\forall n} \tag{7}$$

Hence, the size of the opponent feature vector also depends on the number of bands, scales, and orientations:

$$\begin{aligned} size(\varphi_{x,y}) &= (\binom{B}{2} \times M + B^2 \times (M-1)) \times N = \\ &= size(\psi_{x,y}) + B \times (B-1) \times (M-1) \times N \end{aligned} \tag{8}$$

Note that, in this case, the number of features is considerably increased.

3 Experimental Setup

The hyperspectral image database 92AV3C image has been used in the pixel classification experiments. It was provided by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [13]. The 20-m GSD data was acquired over the Indian Pine Test Site in Northwestern Indiana in 1992. From the original 220 AVIRIS spectral bands our band selection method provides us with ten clusters of bands which are sets of bands that are intended to maximize the information provided [7]. The first cluster contains just one bands, the second contains two bands, and so on.

The experimental activity was held using two filter banks. For the first one, six dyadic scales (the maximum starting from width one and covering all the image) and four orientations were used. For the second one, eight constant frequency bands and four orientations were considered. It has been introduced certain degree of overlapping as recommended in [2]. Gaussian distributions are designed to overlap each other when achieving a value of 0.5.

For each of the scales a classification experiment was held using only the features provided for that scale. In addition, an analysis of the combination of adjacent scales have been performed. In order to study the importance of low frequencies an ascendent joining was performed, characterizing pixels with the data provided by joined ascendent scales. Similarly, the study of the high frequencies was carried out by a descendant joining. Also for medium frequencies, central scales are considered initially and adjacent lower and higher scales are joined gradually.

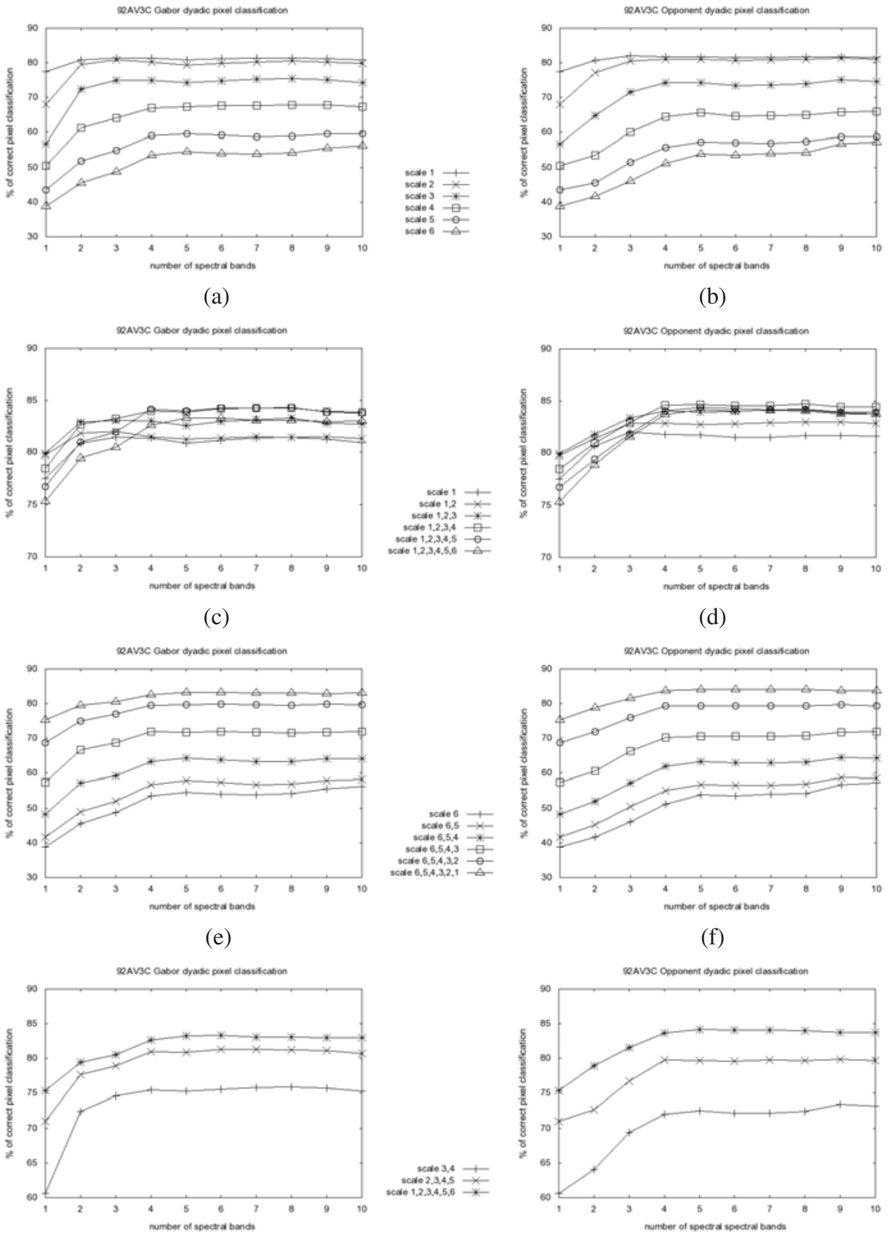
The pixels in the image database are divided in twenty non overlapping sets keeping the a priori probability of each class. Therefore, no redundancies are introduced and each set is a representative set of the bigger original one. Ten classification attempts were carried out for each experiment with the k-nearest neighbor algorithm with $k = 3$ and the mean of the error rates of these attempts was taken as the final performance of the classifier. Each classification attempt uses one of these sets for training and another as test set. Therefore, each set was never used twice in the same experiment.

4 Evaluation of the Results

Figure 1 shows the percentages of correct pixel classification obtained for the experiments that used the dyadic filter bank. Figure 2 shows similar results when the constant filter bank was used.

As it can be observed from both figures, when the characterization processes included all scales, the filter bank using the dyadic tessellation outperforms the constant one. It seems clear that the better the low frequencies are analyzed the better the pixels are characterized. This means that, for this sort of images, the texture information, although still helps in the characterization process, is significantly lower than the information contained in the low frequencies. It can be seen that no scale can ever outperform the classification rates achieved by scale one which achieve up to 81% by itself. In general, the more detail is obtained from low frequencies the best the image is characterized.

For the dyadic tessellation, although scales two and three do not outperform scale one when characterizing independently (Fig. 1a-b), their performance is considerably high. Because the first scales cover a very small part of the frequency domain, the characterization joining scales 1, 2 and 3 improve the pixel classification rates (Fig. 1c-d). In a nutshell, when all (six) scales are used, the classification rates are better than the ones obtained using just the first scale. However, it is worse than the results obtained for the first three scales although having a double number of features. The descendent and central joinings (Figs. 1e-f and 1g-h) clearly show that the performance increases significantly as features derived from lower frequencies are considered.



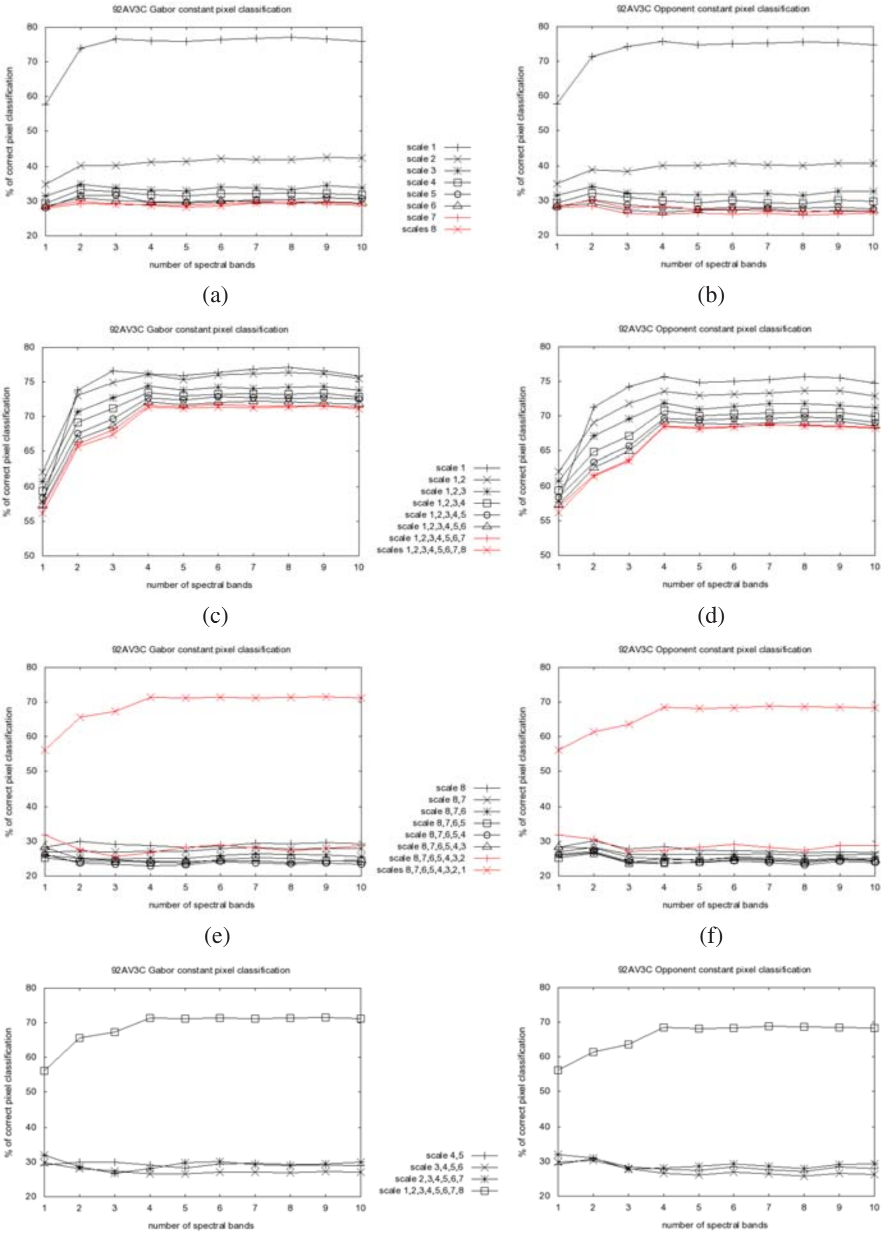
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 1. Pixel classification rates using the filter bank with dyadic tessellation. (a,c,e,g) Gabor features over complex planes (b,d,f,h) Opponent features (a,b) Individual scales (c,d) Ascendent join (e,f) Descendent join (g,h) Central join. Note the different ranges over the Y-axis in each graph.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 2. Pixel classification rates using the filter bank with constant tessellation. (a,c,e,g) Gabor features over complex planes (b,d,f,h) Opponent features (a,b) Individual scales (c,d) Ascendent join (e,f) Descendent join (g,h) Central join. Note the different ranges over the Y-axis in each graph.

Regarding the filter bank, using a constant tessellation (Fig. 2), the first scale is the only one containing discriminant information. This first scale is wide enough in this case to include the information of several scales of the dyadic tessellation. It is very clear from the graphs that the features derived from other scales do not help the characterization processes as the classification rates always decrease. It can be noticed that the best classification rates obtained for the dyadic tessellation is over 84% but is only about 77% for the constant tessellation.

Last but not least, the comparison between the feature extraction methods suggest that opponent features perform similarly to Gabor filters over complex planes. It seems that Gabor features provide better results when using a very small number of spectral bands whereas opponent features provide slightly higher classification rates when more spectral bands are used. Nevertheless, on the whole, the characterization with opponent features requires a larger number of features than Gabor filters, which may worsen performance if a large number of spectral bands is to be considered.

Briefly, spatial analysis between pixels improves hyperspectral satellite images characterization [11] but the nature of this kind of images, which are heterogeneous due to being composed of different homogeneous areas, made low frequencies very important for the characterization task, while texture information may help the process, but not significantly. Furthermore, including much more information but the provided by the low frequency analysis may even decrease the performance because of the so call Hughes phenomenon [5].

5 Conclusions

An analysis of the contribution of each scale to the characterization of hyperspectral images has been performed. As it is known in the texture analysis field, medium and high frequencies play an essential role in texture characterization. However, satellite images cannot be considered as pure texture images since the homogeneity of the different areas in the image is more important than the texture these areas may content. A thoroughly analysis of the contribution of each independent scale and the group composed by low, medium or high frequencies has been carried out. It has been shown that a detailed analysis of low frequencies helps the characterization improving performance. Also a few scales could be considered in the feature extraction process providing by themselves very high classification rates with a few number of features. The comparison between Gabor filters over complex plains and opponent features has shown that the classification rates obtained are almost identical in both cases. The main difference is the number of features required in each case, being much larger for the opponent features.

Acknowledgment

This work has been partly supported by Fundació Caixa Castelló-Bancaixa through grant FPI PREDOC/2007/20 and project P1-1B2007-48, project CSD2007 00018 from Consolider Ingenio 2010, and project AYA2008-05965-C04-04 from Spanish CICYT.

References

1. Chang, T., Kuo, C.C.J.: Texture analysis and classification with tree-structured wavelet transform. IEEE Trans. on Geoscience & Remote Sensing 2, 429–441 (1993)
2. Bianconi, F., Fernández, A.: Evaluation of the effects of Gabor filter parametres on texture classification. Patt. Recogn. 40, 3325–3335 (2007)
3. Jaim, A., Healey, G.: A multiscale representation including oppponent color features for texture recognition. IEEE Trans. Image Process. 7, 124–128 (1998)
4. Shi, M., Healey, G.: Hyperspectral texture recognition using a multiscale opponent representation. IEEE Trans. Geoscience and remote sensing 41, 1090–1095 (2003)
5. Hughes, G.F.: On the mean accuracy of statistical pattern recognizers. IEEE Trans. Inf. Theory 14, 55–63 (1968)
6. Landgrebe, D.A.: Signal Theory Methods in Multispectral Remote Sensing. Wiley, Hoboken (2003)
7. Martínez-Usó, A., Pla, F., Sotoca, J.M., García-Sevilla, P.: Clustering-based Hyperspectral Band selection using Information Measures. IEEE Trans. on Geoscience & Remote Sensing 45(12), 4158–4171 (2007)
8. Mercier, G., Lennon, M.: On the characterization of hyperspectral texture. IEEE Trans. Inf. Theory 14, 2584–2586 (2002)
9. Petrou, M., García-Sevilla, P.: Image Processing: Dealing with Texture. John Wiley and Sons, Chichester (2006)
10. Plaza, A., Martinez, P., Plaza, J., Perez, R.: Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. IEEE Trans. Geoscience and remote sensing 43(3), 466–479 (2005)
11. Rajadell, O., García-Sevilla, P., Pla, F.: Textural features for hyperspectral pixel classification. In: Araujo, H., et al. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 497–504. Springer, Heidelberg (2009)
12. Shaw, G., Manolakis, D.: Signal processing for hyperspectral image explotation. IEEE Signal Process. Mag. 19(1), 12 (2002)
13. Vane, G., Green, R., Chrien, T., Enmark, H., Hansen, E., Porter, W.: The Airborne Visible Infrared Imaging Spectrometer Remote Sens. Environ. 44, 127–143 (1993)

Minimum Variance Gain Nonuniformity Estimation in Infrared Focal Plane Array Sensors

César San-Martin[1] and Gabriel Hermosilla[2]

[1] Information Processing Laboratory, Department of Electrical Engineering,
Universidad de La Frontera. Casilla 54-D Temuco, Chile
csmarti@ufro.cl

[2] Department of Electrical Eng., Universidad de Chile. Casilla 412-3, Santiago, Chile
ghermosi@ing.uchile.cl

Abstract. In this paper, a minimum variance estimator for the gain nonuniformity (NU) in infrared (IR) focal plane array (FPA) imaging system is presented. Recently, we have developed a recursive filter estimator for the offset NU using only the collected scene data, assuming that the offset is a constant in a block of frames where it is estimated. The principal assumption of this scene-based NU correction (NUC) method is that the gain NU is a known constant and does not vary in time. However, in several FPA real systems the gain NU drift is significant. For this reason, in this work we present a gain NU drift estimation based on the offset NU recursive estimation assuming that gain and offset are jointly distributed. The efficacy of this NUC technique is demonstrated by employing several real infrared video se quences.

Keywords: Minimum Variance Estimator, Image Sequence Processing, Infrared Focal Plane Arrays, Signal Processing.

1 Introduction

It is well known that the NU noise in infrared imaging sensors, which is due to píxel-to-pixel variation in the responses of the detector; degrades the quality of IR images [1,2]. In addition, what is worse is that the NU varies slowly on time, depending on the type of technology that is been used. In order to solve this problem, several scene-based NUC techniques have been developed [3,4,5,6,7,8,9]. Scene-based techniques perform the NUC using only the video sequences that are being imaged and not requiring any kind of laboratory calibration technique. In [10] we have developed a recursive filter to estimate the offset of each detector on the FPA. The method is developed using two key assumptions: i) the input irradiance at each detector is a random and uniformly distributed variable in a range that is common in all detectors in FPA; and ii) the FPA technology exhibits important offset non-uniformity with slow temporal drift. The proposed algorithm is developed to operate on one block, short enough to assume that the offset NU can be estimated as a constant in noise

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1047–1053, 2009.

(ECN) [11]. In this paper the gain NU drift is considered and it is obtained from the offset estimated using the ECN method. Afterwards, assuming that the offset and gain are jointly distributed, a minimum variance estimator for the gain can be obtained. This paper is organized as following. In section 2, a review of the ECN NUC method is presented. In Section 3, the gain estimator (GE) proposed is exposed and the results using this method are presented in Section 4. Finally, in Section 5 the conclusions of the paper are summarized.

2 Estimation of a Constant in Noise NUC Method

The pixel-to-pixel variation in the detectors' responses is modeled by the commonly used linear model for each pixel on the infrared focal plane array. For the $(ij)^{th}$ detector, the measured readout signal y_{ij} at a given time n, can be expressed as:

$$y_{ij}(n) = A_{ij}(n)x_{ij}(n) + B_{ij}(n) + v_{ij}(n), \tag{1}$$

where $A_{ij}(n)$ and $B_{ij}(n)$ are the gain and the offset of the $(ij)^{th}$ detector respectively and $x_{ij}(n)$ is the real incident IR photon flux collected by the detector. The term $v_{ij}(n)$ is additive electronic noise represented by a zero-mean Gaussian random variable that is statistically independent of noise in other detectors. The ECN method assumes that $A_{ij}(n)$ is a known constant and $B_{ij}(n)$ remains constant in a block of frames, i.e., $B(n) = B(n-1) = B$, and the model of (1) is re-written as:

$$y(n) = Ax(n) + B + v(n), \tag{2}$$

where the subscript ij is omitted with the understanding that all operations are performed on a pixel by pixel basis. Equation (2) is valid only in a block of frames. The recursive estimator for the offset NU is given by:

$$\hat{B}(n) = C_n \hat{B}(n-1) + K_n y(n), \tag{3}$$

where $\hat{B}(n)$ and $\hat{B}(n-1)$ are the estimates for the offset and C_n and K_n are the coefficients of the filter. Equation (3) can be recursively calculated using the follows equations:

$$C_n = \frac{1 + an}{1 + (n+1)\,a}, \tag{4}$$

$$K_n = \frac{a}{1 + (n+1)\,a}, \tag{5}$$

where a is the convergence control parameter typically less than 1. ECN have two parameters, a and the length of the block of frames, n_b, when the algorithm stops the estimation. The corrected frame is obtained by:

$$\hat{x}(n) = y(n) - \hat{B}(n_b). \tag{6}$$

3 Minimum Variance Gain NU Estimator (GE)

First of all, it is assumed that the gain A and the offset B are Gaussian random processes and they are jointly distributed. Then, the conditional probability of A given B is defined by:

$$p_{A|B}\left(A|B\right) = \frac{p_{A,B}\left(A,B\right)}{p_B\left(B\right)}, \tag{7}$$

where, $p_{A,B}(A,B)$ is the joint probability of A and B, and $p_B B$ is the probability of B (obviously non-zero). After this, a minimum variance estimator for A is formulated from this relationship. With A and B Gaussian random variables jointly distributed, where the information is known with regard to B. Then, the estimator for a minimum variance is defined as only a conditional average of A given B:

$$\hat{A} = E\left[A|B\right] = \int_{-\infty}^{\infty} A p_{A|B}\left(A|B\right) dA, \tag{8}$$

where the error $E\left[\left\|A - \hat{A}\right\|^2 |B\right]$ is minimal. Then, as A and B are assumed Gaussian random process individual and mutually, they are completely defined by their mean and their variance and $p_{A|B}(A|B)$ can be expressed as:

$$p_{A|B}\left(A|B\right) = \frac{\exp\left\{-\frac{\left(A-\left\{\bar{A}+\left(B-\bar{B}\right)\sigma_{AB}\big/\sigma_B^2\right\}\right)^2}{2\left(\sigma_A^2-\sigma_{AB}\sigma_{BA}\big/\sigma_B^2\right)}\right\}}{\sqrt{(2\pi)}\left|\sigma_A^2 - \sigma_{AB}\sigma_{BA}\big/\sigma_B^2\right|^{1/2}}, \tag{9}$$

where, $\bar{A} + \frac{\sigma_{AB}}{\sigma_B^2}\left(B - \bar{B}\right)$ and $\sigma_A^2 - \sigma_{AB}\sigma_{BA}\big/\sigma_B^2$ are the mean and variance respectively. Then, the minimum variance estimator for A is obtained by:

$$\hat{A} = E\left[A|B\right] = \int_{-\infty}^{\infty} A p_{A|B}\left(A|B\right) dA = \bar{A} + \frac{\sigma_{AB}}{\sigma_B^2}\left(B - \bar{B}\right). \tag{10}$$

In our case, we needed to know that $B, \bar{B}, \sigma_B^2$ and σ_{AB}. B were obtained from ECN method when $n = n_b$, the mean and variance of B can be calculated as an approximation to the mean and spatial variance, and the covariance is calculated by:

$$\sigma_{AB} = R_0\sqrt{\sigma_A^2\sigma_B^2}, \tag{11}$$

where, R_0 is the correlation between A and B. This index can be measured from other methods of NUC. Finally, using (11) the GE given by (10) is recast resulting in:

$$\hat{A} = \bar{A} + R_0\frac{\sigma_A}{\sigma_B}\left(B - \bar{B}\right), \tag{12}$$

and the corrected frame is obtained using the following equation :

$$\hat{x}(n) = \frac{y(n) - \hat{B}(n_b)}{\hat{A}}. \tag{13}$$

4 Results

Real IR video data are used to test the ability of the proposed method and reduce NU. The sequence has been collected at 1 PM using a 128 × 128 InSb focal plane array cooled camera Amber Model AE-4128 operating in the 3μm – 5μm range. In the data set, 3000 frames were collected at a rate of 30 frames per second, at 16 bits of resolution. There are data of black bodies radiators, which are used to estimate the gain and the offset associated with each detector. With these parameters, the best correction of nonuniformity is performed, obtaining a sequence that is used as a reference.

As a quantitative measure of performance, the Root Mean Square Error (RMSE) was used, which measures the difference between the reference infrared image and the corrected image using the proposed method. The RMSE is calculated by:

$$RMSE(n) = \sqrt{\frac{1}{pm}\sum_{i=1}^{p}\sum_{j=1}^{m}\left(\hat{x}_{ij}(n) - x_{ij}(n)\right)^2}, \tag{14}$$

where, $p \times m$ is the number of detectors in the focal plane array, $\hat{x}_{ij}(n)$, is the infrared irradiance calculated with the gain estimated by the recursive filter, and $x_{ij}(n)$ is the infrared irradiance calculated by the black-body radiator data. A lower value of RMSE means a good correction of the frame data.

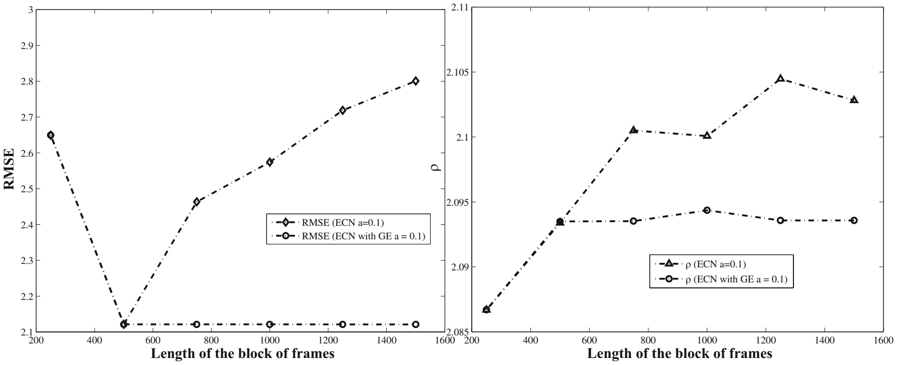
Also the roughness index ρ metric is used to measure performance without reference. ρ delivers information about the level of softness that an image has, i.e., the degree of non-uniformity in this image. This index is calculated by:

$$\rho = \frac{\left\|h * I\right\|_1 + \left\|h^T * I\right\|_1}{\left\|I\right\|_1}, \tag{15}$$

where, the image I is the corrupted or compensated frame, a filter h is needed to find the softness of the image, $*$ represents the convolution and $\|\|_1$ represents the norm L_1. In the same form of RMSE a low value of ρ close to zero indicates a good correction.

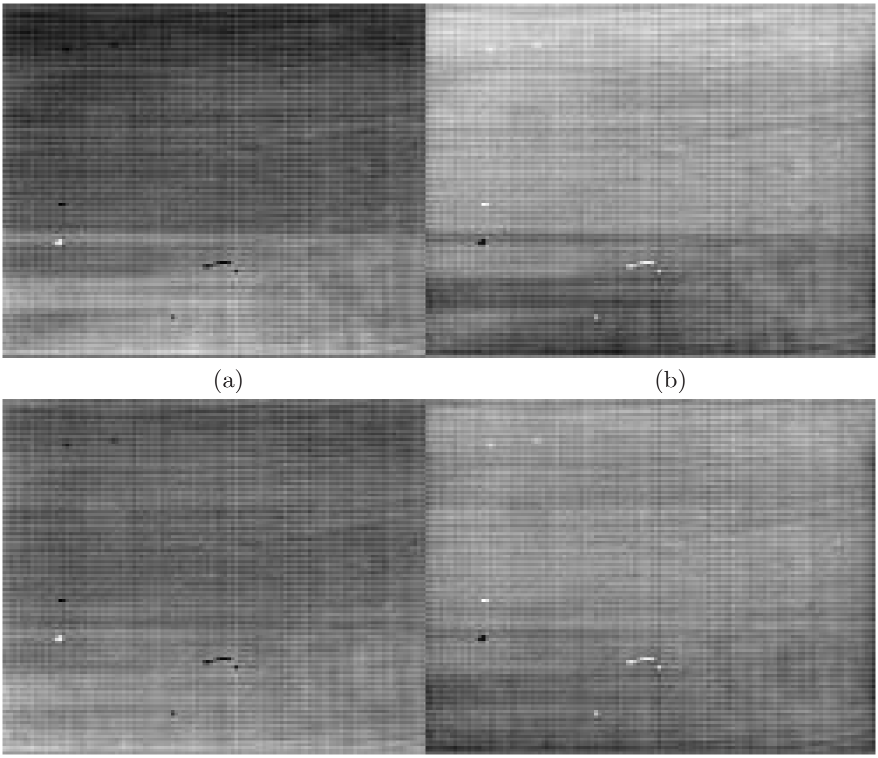
Initially, we estimate the offset NU using the ECN NUC method. From [10], the value for the parameter $a = 0.1$ is the best selection using RMSE and ρ. Then, from the estimated offset the gain NU is obtained using (12), and for each value of $n_b = \{250, 500, 750, 1000, 1250, 1500\}$ the RMSE and ρ are calculated and the results are shown in Fig. 1. The estimated gain NU is presented in Fig. 2b and 2d. For all results we are selected $R_0 = -0.9$ and $\sigma_A = 0.015$.

Finally, a comparison of the performance method is presented in Fig. 3. In this case, the $1600th$ corrupted frame of the real IR sequence is presented (Fig. 3a). The ECN compensates the corrupted frame and the results are shown in Fig. 3b, and the GE NUC method generates the corrected frame in Fig. 3c. The corresponding RMSE and ρ values are presented in Table 1. For this case, the RMSE values are 3.37, 2.96 and 2.12 for the corrupted frame, corrected frame with ENC and corrected frame with GE, respectively. The ρ values correspond to 2.417 for the corrupted frame, 2.108 for the compensated frame using ENC



(a) (b)

Fig. 1. RMSE (a) and ρ (b) for the compensated frame using ECN and ECN with GE NUC method. In this case, $n_b = \{250, 500, 750, 1000, 1250, 1500\}$ and $a = 0.1$.



(a) (b)

(c) (d)

Fig. 2. Different offset and gain NU estimated using the proposed method. a) and c) correspond to the estimation of B using ECN for $n_b = 500$ and $n_b = 1000$ respectively; b) and d) show the gain NU using GE method for $n_b = 500$ and $n_b = 1000$ respectively.

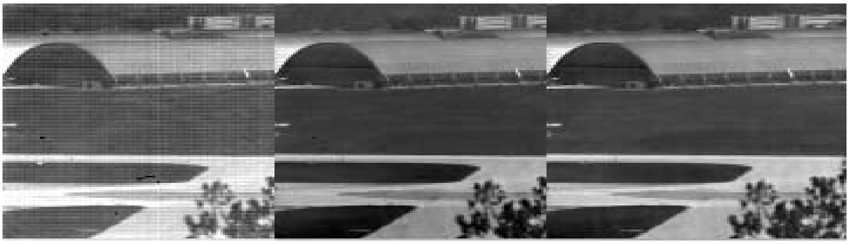


Fig. 3. Performance of the proposed method using real IR data. (a) corrupted frame, (b) compensated frame using ECN method, and (c) corrected frame using ECN with GE method.

Table 1. The calculated RMSE and ρ parameters for real IR frames corrected for NU by using the ECN method and ECN with GE method

Frame	RMSE	ρ
Corrupted	3.37	2.417
Corrected using ECN	2.96	2.108
Corrected using ECN with GE	2.12	2.093

and 2.093 for ECN with GE. From these results clearly the proposed method generates a better performance when the gain NU drift is considered.

5 Conclusions

A recursive estimation for gain NU on infrared imaging systems is proposed in this paper. It was shown experimentally using real IR data that the method is able to reduce non-uniformity substantially. Indeed, the method has shown an acceptable reduction of nonuniformity after processing only approximately 500 frames. The main advantage of the method is a simplicity using only fundamental estimation theory. The key assumption of the proposed method is that the offset and the gain are jointly distributed. The offset is estimated using a recursive filter, and then, the gain is obtained using a minimum variance estimator. The results presented showed that this assumption is validated with real IR data.

Acknowledgments

This work was partially supported by Universidad de La Frontera, Proyecto DIUFRO 08-0048. The authors wish to thank Ernest E. Armstrong (OptiMetrics Inc., USA) for collecting the data, and the United States Air Force Research Laboratory, Ohio, USA.

References

1. Milton, A., Barone, F., Kruer, M.: Influence of nonuniformity on infrared focal plane array performance. Optical Engineering 24, 855–862 (1985)
2. Mooney, J., Shepherd, F., Ewing, W., Murguia, J., Silverman, J.: Responsivity nonuniformity limited performance of infrared staring cameras. Optical Engineering 28, 1151–1161 (1989)
3. Harris, J., Chiang, Y.: Nonuniformity correction of infrared image sequences using constant statistics constraint. IEEE Trans. on Image Processing 8, 1148–1151 (1999)
4. Hayat, M., Torres, S., Amstrong, E., Cain, S., Yasuda, B.: Statistical algorithm fo nonuniformity correction in focal plane arrays. Applied Optics 38, 773–780 (1999)
5. Averbuch, A., Liron, G., Bobrovsky, B.: Scene based non-uniformity correction in thermal images using Kalman filter. Image and Vision Computing 25, 833–851 (2007)
6. Scribner, D., Sarkady, K., Kruer, M.: Adaptive nonuniformity correction for infrared focal plane arrays using neural networks. In: Proceeding of SPIE, vol. 1541, pp. 100–109 (1991)
7. Scribner, D., Sarkady, K., Kruer, M.: Adaptive retina-like preprocessing for imaging detector arrays. In: Proceeding of the IEEE International Conference on Neural Networks, vol. 3, pp. 1955–1960 (1993)
8. Torres, S., Vera, E., Reeves, R., Sobarzo, S.: Adaptive scene-based nonuniformity correction method for infrared focal plane arrays. In: Proceeding of SPIE, vol. 5076, pp. 130–139 (2003)
9. Torres, S., Hayat, M.: Kalman filtering for adaptive nonuniformity correction in infrared focal plane arrays. The JOSA-A Opt. Soc. of America 20, 470–480 (2003)
10. Martin, C.S., Torres, S., Pezoa, J.E.: Statistical recursive filtering for offset nonuniformity estimation in infrared focal-plane-array sensors, in press Infrared Physics & Technology (2008)
11. Poor, H.V.: An introduction to signal detection and estimation, 2nd edn. Springer, New York (1998)

Movement Detection and Tracking Using Video Frames

Josue Hernandez[1], Hiroshi Morita[2], Mariko Nakano-Miytake[1],
and Hector Perez-Meana[1]

[1] National Polytechnic Institute, Av. Santa 1000, 04430 Mexico D. F. Mexico
[2] The University of Electro-Communications, 182-8585, Tokyo, Japan
mariko@infinitum.com.mx, hmperezm@ipn.mx

Abstract. The use of image processing schemes as part of the security systems have been increasing, to detect, classify as well as to tract object and human motion with a high precision. To this end several approaches have been proposed during the last decades using image processing techniques, because computer vision let us to manipulated digital image sequences to extract useful information contained in a video stream. In this paper we present a motion detection algorithm using the movement vectors estimation which are subsequently filtered to obtain better information about real motion into a given scenes. Experimental results show that the accuracy of proposed system.

Keywords: Motion Vectors, Movement Detection, Surveillance System, Surveillance system development.

1 Introduction

The advance of electronic and computer technologies increase the ability to perform video analysis to extract useful information from a video frames to carry out [1] motion detection and characterization [2], remote sensing, and pattern recognition, among others [3]-[5]. Pattern recognition is a research area that has been amply studied during the last years, it use information contained in video sequences; In many applications of pattern recognition, approaches with recognition capability are usually based on a corpus of data which is treated either in a holistic manner or which is partitioned by application of prior knowledge [5] like shape, velocity, direction, texture, magnitude, behavior and so on from different kind of objects around the area of interest.

Several approaches have been proposed to solve the problem of detecting and tracking motion during the last several years [6]-[16]. Some of them are describe in the following paragraphs [6]. Reference [7] proposes a motion detection approach based on the MPEG image compression algorithm in which the estimation of detection motion and the moving object direction only use the information contained in the MPEG motion vectors and the DC coefficients of the DCT directly extracted from the MPEG bit stream of the processed video. Evaluation results of this method shows that it can handle several situations where moving objects are present in the scene observed with a mobile camera. However, the

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1054–1061, 2009.

efficiency of the moving object detection depends on the quality of MPEG motion vectors [8-12]. This method also has a constraint in the context of contend-based video browsing and indexing that should be taken into account when the MPEG encoder is selected. Reference [9] proposed an algorithm that intends to extend the capabilities to MPEG2 streams, allowing tracking objects selected by a user throughout the sequence. The way in which the tracking has been realized, is through exploitation of the motion information already present in the bit stream and generated by the decoder, another important feature of this scheme is that it is not necessary to decode the full sequence, if the user wants to access only part of the video stream [10]. This proposed scheme performs well in assisting the information insertion/retrieval process. However, no segmentation or filtering techniques are used for the extraction and tracking of the objects, since it relies exclusively on the motion information already provided by the MPEG encoder. Evaluation results show that the algorithm performs well though it is slightly dependent on the object shape. Hariharakrishnan et al [8] proposes an algorithm in which the tracking is achieved by predicting the object boundary using motion vectors, followed by contour update, using occlusion/disocclusion detection. An adaptive block-based approach has been used for estimating motion between consecutive frames. Here an efficient modulation scheme is used to control the gap between frames used for object tracking. The algorithm for detecting occlusion proceeds in two steps. First, covered regions are estimated from the displaced frame difference. Next these covered regions are classified into actual occlusions and false alarms using the motion characteristics. Disocclusion detection is also performed in a similar manner [8].

This paper proposes an algorithm for detection and tracking of movement of objects and persons based on a video sequence processing. Evaluation results show that proposed scheme provides a fairly good performance when required to detect relevant movements and tracking the motion of objects under analysis.

2 Proposed Movement and Tracking Detection System

The proposed system, firstly estimates the motion vectors, using an input video frames, which are then filtered to reduce distortion due to noise and deficient illumination. Next using the estimated motion vectors the movement trace is estimated to determine if it is a relevant or irrelevant motion. Finally if the movement is relevant, its tracking is carried out until the object left the restricted zone.

2.1 Motion Vector Estimation

The motion estimation is carried out dividing, firstly, the actual image at time, t-1, into a non-overlapping macro-blocks of 16x16 pixels. Next in the image frame at time t, the algorithm looks for the region that closely matches the macro-block under analysis [2]. Taking in account that the time difference in time between two consecutive images in a given frame is relatively small, only is necessary to carry out the analysis in a region slightly larger than the given macro-block. Here the distance and direction that minimizes the criterion given by [5]

$$MAE(i,j)=\frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\left|C(x+k,y+l)-R(x+i+k,y+j+l)\right| \tag{1}$$

Where *C(x+k,y+l)* is the *(k,l)th* píxel of macro-block *(x,y)* of the actual frame and *R(x+i+k.y+j+l*$^{)}$ ^{i}s the macro-block in the position *(x+i,y+j)* in the reference frame, where $-p \le i \le p$ $y-p \le j \le p$. Then the vector motion magnitude and direction is obtained from to actual position (x,y) in the actual frame, to the position (x+i,y+j) in which the MAE (i,j) is minimum. This process is illustrated in Fig. 1.

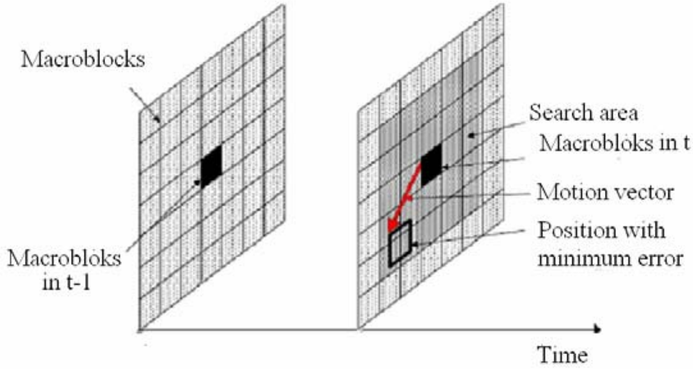


Fig. 1. Motion vector estimation process

Several methods have been proposed for motion vector estimation, which are based on the MAE (Mean Absolute error) minimization (1), all of them providing similar results although their computational complexity presents important differences. Among the Hierarchical estimation method presents the less computational complexity and then is one of the most widely used motion vector estimation method. Here to reduce the computational complexity, the image is decimated by 2 using low pass filters, reducing in such way the image size. Thus the position *(x,y)* in the original image becomes the position *(x/2,y/2)* in the image corresponding to the first decomposition level. Subsequently a second decomposition level is applied using a low pass filter, such that the original *(x,y)* point becomes the point *(x/4,y/4)* in the second decomposition level image [5].

After the second decomposition level is performed, the motion vector estimation starts in the second decomposition level, in which the image size is 1/16 of the original image size, with a macro-block of size 4x4. Assuming that the point *(u_2,v_2)* corresponds to the minimum MAE of the actual macro-block, the search of the motion vector in the decomposition level 1 is carried out in the macro-blocks of 8x8 whose search start in *(x/2+2u_2,y/2+2v_2)*, with a search range of [-1,1] around the origin pixel. Finally the motion vector estimation in the level decomposition 0 is carried out with macro-blocks of size 16x16, with search starting at point /(x+2u_1,y+2v_1) with a search range equal to [-1,1], around the origin pixel. Here

(u_1,v_1) corresponds to the minimum MAE at the level decomposition 1. These motion vectors consist of horizontal and vertical components which are used to estimate the motion vector magnitude and angle. Finally using these data it is possible to determine in the movement is relevant or not. Figure 2 illustrate this process.

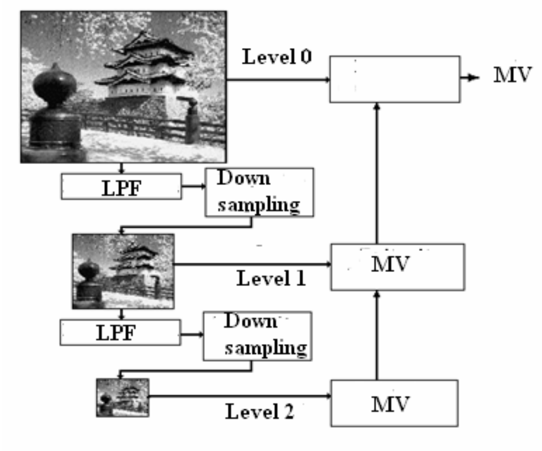


Fig. 2. Hierarchical motion vector estimation algorithm

Several times the estimated motion vectors only represent abrupt illumination changes in the image or small vibrations of the camera used to capture the image sequence, resulting in wrong movement estimation if these distortions are not cancel or at least reduced. To this end, the estimated motion vectors are filtered using a one dimensional median filter as shown in Fig. 3, before the overall movement estimation. This allows to cancel most motion vectors that no provides useful information, such as those due to illumination changes, background movement, etc. This process is shown in Fig. 3.

Because the amount of noisy motion vectors is relatively small in comparison with the correctly estimated motion vectors, as well due to the fact that the noisy vectors are continuous ore similar among them, as happen with the actual motion vectors, the noisy motion vectors can be easily distinguish form the remaining ones. On the other hand, because the motion vectors are estimated one by one it is not necessary to create a temporal register, to be use during the filtering process [5]. Thus after the noisy vectors have been eliminated, the information generated by objects moving in no relevant directions is cancelled, based on the fact that movement direction is different to that of the security zone. This fact allows the proposed algorithm to measure the relevance level, according to their position in the scene. Thus a motion vector is eliminated if its position does not change after several video sequences or if the motion direction is opposite the restricted zone, as shown in Fig, 4.

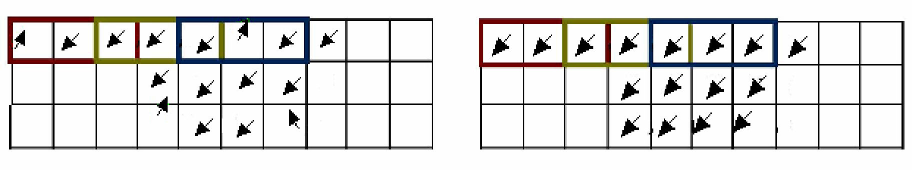


Fig. 3. Motion vectors filtering process

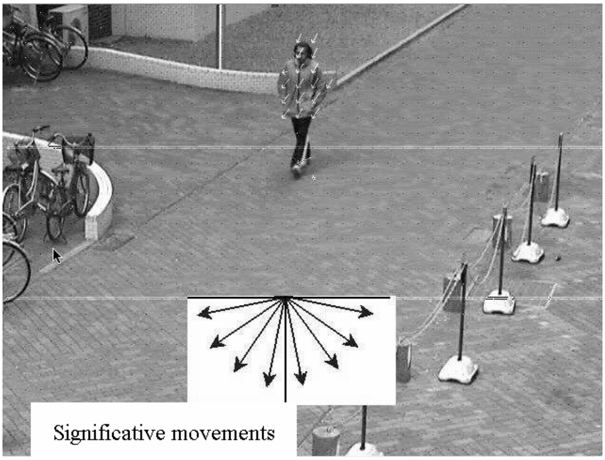


Fig. 4. Relevant movement detection in the direction of the restricted zone

2.2 Trajectory Estimation

Once the motion vectors have been estimated, it is necessary to estimate other important parameters such as: the speed, direction and movement tracking of the moving object present in the video sequence. To this en firstly it can be used the information provided by the motion vector to estimate the movement angle to classify the movement in, either, relevant or not relevant. To this end, we can take in account that a relevant movement is that in which the people enter or intend to enter into the restricted zone. This fact implies that, according with the video camera position, the motion vectors angles must be between 200 and 340 degrees. This fact takes in account that a movement in not relevant if it takes place inside the restricted zone, or if its direction is from inside to outside the restricted zone.

Other important factor that must be consider to obtain an accurate estimation of the motion vectors is the camera position, because it allows to obtain constant motion vectors, with similar magnitude during all trajectory, avoiding detection errors. This fact also allows that a person magnitude may be represented using among 9 to 13 motion vectors. A correct distance also allows adding some divisions to measure the movement importance and be able to track the position of a given object during a determined time; as well as to add several counters to determine the number of

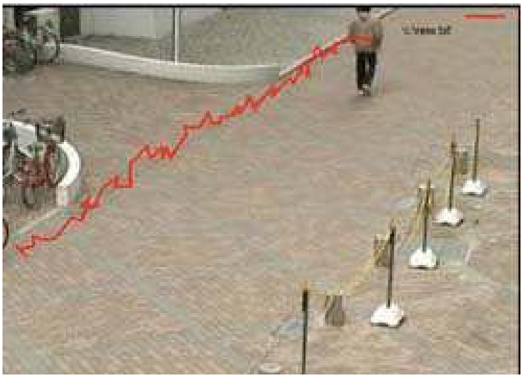


Fig. 5. Tracking ability of proposed algorithm

persons that are generating relevant movements, as well as the direction of these movements.

2.3 Movement Tracking

Other important issue is the tracking of the generated relevant movement, which means that not only is important to determine if the movement a relevant movement has been produced, but also to have the ability to track the movement of the person generating it until the go out of the restricted zone. The min difficulty to track the object movement using a sequence of successive images is the object localization in the image sequences, especially when the movement of them is relatively faster than the images rate. For these reason, the tracking systems usually uses moving models that intend to predict the variations of the image due to possible movements present in the object trajectory. Although these methods provide fairly good results, they may be computationally complex. Thus to reduce the computational complexity, the object is divided in NXM non overlap blocks. Next using the motion vector estimation described above, the movement of each corner is estimated. Once the motion vectors are obtained, the tracking of object motion in each frame is obtained as the resultant vector of all individual motion vectors in such frame. Finally to obtain a smoother trajectory the resulting vector in the previous step is filtered using a low pass filter.

3 Evaluation Results

The proposed system was evaluated using computer simulation using a Power Mac G4, with a data bus speed of 167MHz, a CPU of 1.25GHz ad 1.25GB of RAM memory. The video sequences used to evaluate the proposed algorithm were previously recorded with a resolution of 640x480 pixels per image. The obtained results show that the proposed algorithm is able to accurately detect the relevant movements and correctly tract the person u object motion.

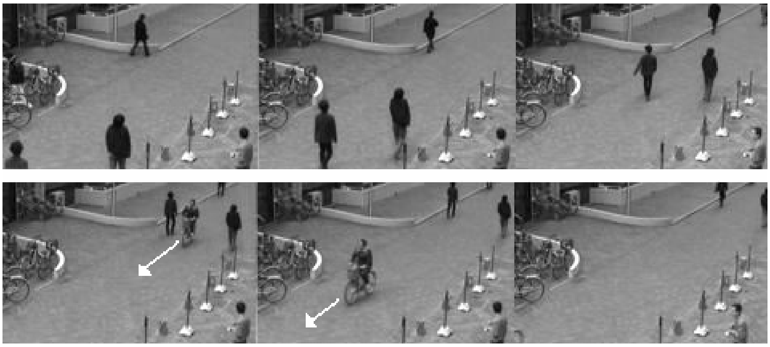


Fig. 6. Detection performance of proposed algorithm, here in the fourth and fifth image there are a relevant movement

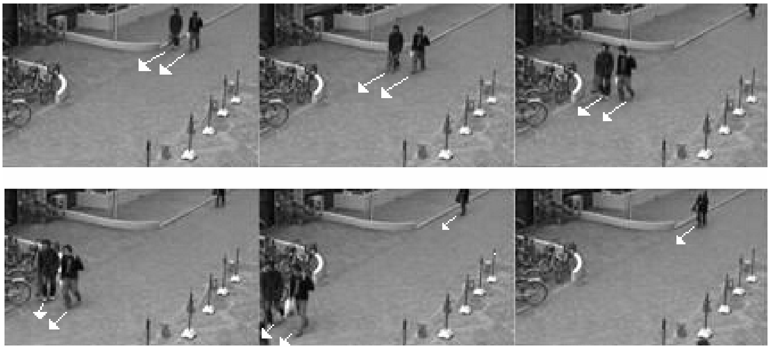


Fig. 7. Two persons moving to the restricted zone

4 Conclusions

This paper proposed an automatic movement detection system, using a video sequence, based on the motion vector estimation, which are filtered to eliminate the noisy and distorted vectors, due to illumination variations and background movement. Proposed algorithm is also to discriminate between relevant and non relevant movements which allows only take in account the movements whose direction is from outside to inside the restricted zone. Using also a motion vector estimation the algorithm is also able to tract the trajectory of a given person whose movement is from inside to outside of the restricted zone.

Acknowledgements

The authors thanks the National Science and Technology Council, to The Science and Technology Institute of the D. F. Mexico, and to the National Polytechnic Institute of Mexico for the financial support provide during the realization of this research.

References

1. Jones, B.T.: Low-Cost Outdoor Video Motion and Non-Motion Detection. Processing of Security Technology, 376–380 (1995)
2. Paladino, V.: Introduction of Video Compression Under the Standard MPEG-2. The Institute of Electronic Engineer,Spain, 3–24 (2005)
3. Richardson, I.E.: H.264 and MPEG-4 Video Compression, Video Coding for the Next-Generation Multimedia, pp. 27–41. Wiley, UK (2004)
4. Watkinson, J.: The MPEG Handbook: MPEG-1, MPEG-2, MPEG-4. Focal Press (2001)
5. Tudor, P.N.: MPEG-2 Video Compression: Tutorial. Journal of Electronics and Communication Engineering, 1-5 (December 1995)
6. Zhang, Z.: Mining Surveillance Video for Independent Motion Detection. In: IEEE Internacional Conference on Data Mining, pp. 741–744 (2005)
7. Favalli, L., Mecocci, A., Moschetti, F.: Object Tracking For Retrieval in MPEG2. IEEE, Trans. on Circuit and Syst. for Video Technology, 427–432 (2000)
8. Hariharakrishnan, K., Schonfeld, D., Raffy, P., Yassa, F.: Video Tracking Using Block Matching. In: IEEE, International Conference on Image Processing, pp. 945–948 (2003)
9. Yoneyama, A., Nakajima, Y., Yanagihara, H., Sugano, M.: Moving Object Detection from MPEG Video Stream. Systems and Computers in Japan 30(13), 1–11 (1999)
10. Avidan, S.: Support Vector Tracking. IEEE 26(8), 1064–1071 (2004)
11. Nguyen, H., Smeulders, A.: Fast Occluded Object Tracking by a Robust Appearance Filter. IEEE Trans. on Image Processing 26(8), 1099–1103 (2004)
12. Lin, M., Tomasi, C.: Surfaces with Occlusions from Layered Stereo. IEEE 26(8), 1073–1098 (2004)
13. Sebastian, T., Klein, P., Kimia, B.: Recognition of Shapes by Editing Their Sock Graphs. IEEE Trans. on Image Processing 26(5), 550–554 (2004)
14. Coding Audiovisual Objects Part 2, International Standard Organization / Int. Electronics Communications (ISO/IEC), 14496
15. Hernández García, J., Pérez-Meana, H., Nakano Miyatake, M.: Video Motion Detection using the Algorithm of Discrimination and Hamming Distance. In: Ramos, F.F., Larios Rosillo, V., Unger, H. (eds.) ISSADS 2005. LNCS, vol. 3563, pp. 321–330. Springer, Heidelberg (2005)
16. Gryn, J.M., Wlides, R., Tsotsos, J.K.: Detecting Motion Patterns via Directional Maps with Applications to Surveillance. Computer Vision and Image Understanding 113, 291–307 (2009)

A New Steganography Based on χ^2 Technic

Zainab Famili[1], Karim Faez[2], and Abbas Fadavi[3]

[1] Department of Electrical, Computer and IT Eng., Azad University, Qazvin, Iran
z_electron590@yahoo.com
[2] Department of Electrical Eng., Amirkabir University of Tech, Tehran, Iran
kfaez@aut.ac.ir
[3] Department of Electrical Eng., Azad University, Garmsar, Iran
abbas_fadavi@yahoo.com

Abstract. In this paper, we proposed a new method for Steganography based on deceiving χ^2 algorithm. Since the cover image coefficients and stego image coefficients histograms have sensible difference for purposes of statistical properties, statistical analysis of χ^2-test reveals the existence of hidden messages inside stego image .We are introducing the idea for hiding messages in the cover image. It causes that DCT (Discrete Cosine Transforms) coefficient histogram not having remarkable modification before and after embedding message. As a result, the identifying of hidden message inside an image is impossible for an eavesdropper through χ^2-test. In fact, the proposed method increases the Steganography security against χ^2-test, but the capacity of embedding messages decreases to half.

Keywords: Steganography, cover image, stego image, χ^2-test.

1 Introduction

Information hiding is a recently developed technique in the information security field and has received significant attention from both industry and academia [1]. Steganography is one technique for hiding information with heavy application in military, diplomatic, and personal area [2]. In the past, people used hidden tattoos or invisible ink to convey Steganographic contents. Today, computer and network technologies provide easy-to-use communication channels for Steganography [3,4]. In today's digital world, invisible ink and paper have been replaced by much more versatile and practical covers for hiding messages inside media such as digital documents, images ,video, and audio files. The digital image is one of the most popular digital mediums for carrying covert messages. There are two main branches Steganography and digital watermarking [1]; the modifications are in spatial domain for the watermarking, and in the frequency domain for the Steganography. The information-hiding process in a Steganographic system begins by identifying a cover medium's redundant bits (bits which can be modified without destroying that medium's integrity) [5].Then this redundant bits are replaced with the data by the hidden messages. In space-hiding systems, one simple method is that of least significant bit Steganography or LSB

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1062–1069, 2009.

embedding. LSB embedding has the merit of simplicity, but suffers from a lack of robustness, and it is easily detectable [6,7]. Steganography goal is to keep hidden message inside an image undetectable, but Steganographic systems for the reason of their invasive nature leave detectable traces in the statistical properties cover medium. Modifying the cover medium changes its statistical properties, so eavesdroppers can detect the distortions in the resulting stego medium's statistical properties. To accommodate a secret message, the original image, also called the cover-image, is slightly modified by the embedding algorithm. As a result, the stego-image is obtained [8,9]. Each Steganographic communication system consists of an embedding algorithm and an extraction algorithm. The system is secure if the stego images do not contain any detectable artifacts due to message embedding. It means the stego images should have the same statistical properties as the cover images [10]. Three different aspects in information-hiding systems contend with each other: capacity, security, and robustness [11]. Capacity refers to the amount of information which can be hidden in the cover medium, security refers to an eavesdropper's inability to detect hidden information, and robustness refers to the amount of modification the stego medium can withstand before an adversary can destroy the hidden information [3]. Steganography strives for high security and capacity, which often entails that the hidden information is fragile. While digital watermarking is mainly used for copyright protection of electronic products [12,13,14] and its primary goal is to achieve a high level of robustness. For Steganography to remain undetected, the unmodified cover medium must be kept secret, because if it is exposed, a comparison between the cover and stego media immediately reveals the changes. The plan of this paper is given by a brief review of JPEG image format in Section2. In Section 3, we present Steganographic systems. After reviewing statistical analysis in Section4, we present out proposed method in Section 5. In Section 6 we summarized the results.

2 JPEG Image Format

The format of cover-image is important because it significantly influences the design of the stego system and its security. There are many advantages using images in JPEG format as carrier-image in steganographic applications. JPEG [15] is a popular and widely-used image file format and has become a de facto standard for network image transmission. If we apply JPEG (Joint Photographic Experts Group) images for data hiding; the stego-image will draw less attention of suspect than that with most other formats. JPEG format operates in a DCT transform space and is not affected by visual attacks [16]. The JPEG image format uses a discrete cosine transform (DCT) to transform successive 8×8 pixel blocks of the image into 64 DCT coefficients each. The DCT coefficients $F(u,v)$ of an 8×8 block of image pixels $f(x,y)$ are given by equation (1):

$$F(u,v) = \alpha(u) \cdot \alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos[\frac{(2x+1)u\pi}{16}] \cos[\frac{(2y+1)v\pi}{16}]. \qquad (1)$$

Afterwards, the equation (2) quantizes the coefficients:

$$F^Q(u,v) = round\frac{F(u,v)}{Q(u,v)}. \tag{2}$$

$Q(u,v)$, is a 64-element quantization table. (This table is given in reference [18]). We can use the least-significant bits of the quantized DCT coefficients, for which $F^Q(u,v) \neq 0$ and $\neq 1$, are used as redundant bits into which the hidden message is embedded [17]. For more information about JPEG, the reader is referred to [18].

3 Steganographic Systems

There are five popular Steganographic algorithms which hide information in JPEG images [19]:

- JSteg: Its embedding algorithm sequentially replaces the least-significant bit of DCT coefficients with the message's data. The algorithm does not require a shared secret; as a result, anyone who knows the Steganographic system can retrieve the message hidden by JSteg [1].
- JSteg-Shell: It compresses the image contents before embedding the data with JSteg. JSteg-Shell uses the stream cipher RC4 (Ron's code #4 or Rivets) for encryption [20].
- JPHide: Before the content is embedded, it is Blowfish [21], encrypted with a user-supplied pass phrase.
- Outguess: Outguess 0.1 is a Steganographic system which improves the encoding step by using a PRNG to select DCT coefficients at random, and Outguess 0.2, which includes the ability to preserve statistical properties [22].
- F5 algorithm: In F5 instead of replacing the least-significant bit of a DCT coefficient with message data, it decrements the absolute value of DCT coefficients in a process called matrix encoding. As a result, there is no coupling of any fixed pair of DCT coefficients [10,23].

4 Statistical Analysis

Statistical tests can reveal if an image has been modified by Steganography by testing whether an image's statistical properties deviate from a norm. Westfield and Pfitzmann observe that for a given image, the embedding of encrypted data changes the histogram of its color frequencies [19]. In the following, we clarify their approach and show how it applies to the JPEG format. In their case, the embedding process changes the least signification bits of the colors in an image. The colors are addressed by their indices in the color table. If n_i and n_i^* are the frequencies of the color indices before and after the embedding respectively, then the following relation is likely to hold

$$|n_{2i} - n_{2i+1}| \geq |n_{2i}^* - n_{2i+1}^*|. \tag{3}$$

In other words, the embedding algorithm reduces the frequency difference between adjacent colors. In an encrypted message, zeros and ones are equally distributed. Given uniformly distributed message bits, if $n_{2i} > n_{2i+1}$, then pixels with color 2i are changed more frequently to color 2i + 1 than the pixels with color 2i + 1 are changed to color 2i. The same is true in the case of the JPEG data format. Instead of measuring the color frequencies, we observe differences in the frequency of the DCT coefficient. Figure (1) displays the histogram before and after a hidden message has been embedded in a JPEG image [3,22]. A χ^2 - test used to determine whether the observed frequency distribution y_i in the image matches a distribution which shows distortion from embedding hidden data [4]. Although we do not know the cover image, we know that the sum of adjacent DCT coefficients remains invariant, which lets us compute the expected distribution y_i^* from the stego image .We then take the arithmetic mean,

$$y_i^* = \frac{n_{2i} + n_{2i+1}}{2}. \tag{4}$$

To determine the expected distribution and compare it against the observed distribution

$$y_i = n_{2i}. \tag{5}$$

The χ^2 value for the difference between the distributions is given as

$$\chi^2 = \sum_{i=1}^{\nu+1} [\frac{(y_i - y_i^*)^2}{y_i^*}]. \tag{6}$$

Where ν are the degrees of freedom that is, one less than the number of different categories in the histogram [3].

5 Proposed Method

Indeed, after doing Steganography inside an image, χ^2 algorithm is operated on the basis of sensible modification which will increase the difference between Δn, and Δn^*. n_{2i} is DCT coefficient frequency in $2i$ before of the embedding messages,and n_{2i}^* is DCT coefficient frequency in $2i$ after of the embedding messages.

$$\Delta n = n_{2i} - n_{2i+1}, \qquad \Delta n^* = n_{2i}^* - n_{2i+1}^* \ . \tag{7}$$

We propose a new method in this article to endure that the differences between n_{2i} and n_{2i+1} don't have remarkable changes before and after embedding message. In this approach, two sequential DCT coefficients hide only one message bit. In this method, the hiding capacity is reduced to half as compared with JSteg. At first, we arrange DCT coefficients to (2i, 2i+1) groups in terms of i (see table 1).

In study of each group, we will realize that with LSB modification, each members of group, changes its own group and no member of one group conveys to

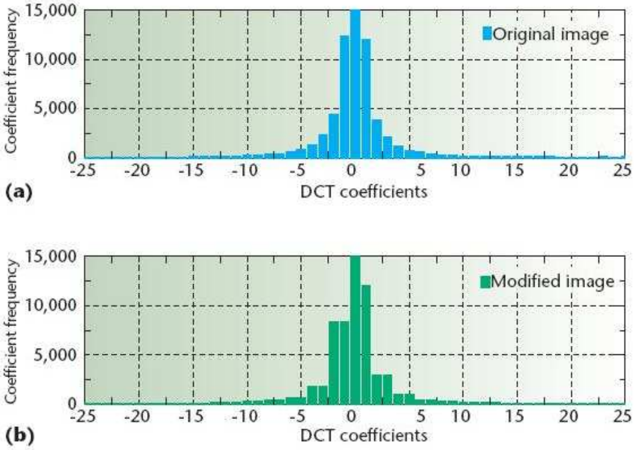


Fig. 1. Frequency histograms. Sequential changes to the (a) original and (b) modified image's least-sequential bit of discrete cosine transform coefficients tend to equalize the frequency of adjacent DCT coefficients in the histograms [3].

Table 1. Grouping the of two adjacent DCT coefficients according to i

	Group1	Group2	Group3	Group4	.....
2i	2	4	6	8	.....
2i+1	3	5	7	9	.....

Table 2. This table indicate the new coefficient replacement according to the message bit to be 0 or 1

Message Bit	New Coefficient
0	(2i,2i) or (2i+1,2i)
1	(2i,2i+1) or (2i+1,2i+1)

other group. In JSteg method, applying Steganography algorithm and transferring the coefficients in each group changes the difference between Δn, and Δn^*in group. So eavesdroppers will be able to recognize the existence of message. To avoid of this subject, we introduce a new approach to minimize the difference between Δn, Δn^*. First of all, we obtain in each group, and we examine the coefficients of each group separately two by two. According to the value of two coefficients and the hiding message, Δn and Δn^* two coefficients are replaced by two new coefficients. With due attention to a message bit to be 0, or 1, we create the table (2) for new coefficients. These two coefficients are chosen optional for hiding message.

For example, we assume i be equal to 4 ($i = 4$), and two sequential coefficients equal to (9, 8). For the purpose of a hiding message bit with zero value, we can replace (8, 8) or (9, 8) according to table (2). There are three cases in choosing (8, 8) or (9, 8) in the example.

Table 3. General convert method

Old Coefficient1	Old Coefficient2	message	Δn and Δn^*	New Coefficient1	New Coefficient2
2i	2i	0	$\Delta n > \Delta n^*$	2i	2i
2i	2i	0	$\Delta n = \Delta n^*$	2i	2i
2i	2i	0	$\Delta n < \Delta n^*$	2i+1	2i
2i	2i	1	$\Delta n > \Delta n^*$	2i	2i+1
2i	2i	1	$\Delta n = \Delta n^*$	2i	2i+1
2i	2i	1	$\Delta n < \Delta n^*$	2i	2i+1
2i	2i+1	0	$\Delta n > \Delta n^*$	2i	2i
2i	2i+1	0	$\Delta n = \Delta n^*$	2i+1	2i
2i	2i+1	0	$\Delta n < \Delta n^*$	2i+1	2i
2i	2i+1	1	$\Delta n > \Delta n^*$	2i	2i+1
2i	2i+1	1	$\Delta n = \Delta n^*$	2i	2i+1
2i	2i+1	1	$\Delta n < \Delta n^*$	2i+1	2i+1
2i+1	2i	0	$\Delta n > \Delta n^*$	2i	2i
2i+1	2i	0	$\Delta n = \Delta n^*$	2i+1	2i
2i+1	2i	0	$\Delta n < \Delta n^*$	2i+1	2i
2i+1	2i	1	$\Delta n > \Delta n^*$	2i	2i+1
2i+1	2i	1	$\Delta n = \Delta n^*$	2i	2i+1
2i+1	2i	1	$\Delta n < \Delta n^*$	2i+1	2i+1
2i+1	2i+1	0	$\Delta n > \Delta n^*$	2i+1	2i
2i+1	2i+1	0	$\Delta n = \Delta n^*$	2i+1	2i
2i+1	2i+1	0	$\Delta n < \Delta n^*$	2i+1	2i
2i+1	2i+1	1	$\Delta n > \Delta n^*$	2i	2i+1
2i+1	2i+1	1	$\Delta n = \Delta n^*$	2i+1	2i+1
2i+1	2i+1	1	$\Delta n < \Delta n^*$	2i+1	2i+1

Table 4. Result of running χ^2-test over cover image, stego image (JSteg method), Stego image (proposed method)

	$\chi^2$for cover image	$\chi^2$for stego image(JSteg method)	χ^2 for stego image(proposed method)
1	140	10	136
2	329	10	316
3	611	11	508
4	360	11	340
5	511	14	463
6	227	14	197
7	245	14	223

1-If $\Delta n > \Delta n^*$, it means that the frequency of eights is less than its frequency in the cover image. So that, it should be convert one 9 to one 8.There for, we use (8, 8). As a result, one occurrence of the nines is decreased and it is increased to 8.

2 - If Δn is Δn^*, it means that we don't need to change the number of eights and nines, then new coefficients are same (9, 8).

3 - If $\Delta n < \Delta n^*$,it means that the frequency of eights is more than its frequency in the cover image. Thus, it should convert one 8 to one 9. Because

in transmitting this message, there isn't any (9, 9) then we use (9, 8).Using this step stop the difference between Δn , and Δn^* to get higher. We show general convert method (approach) in table (3).

6 Conclusion and Result

Our proposed algorithm is applied on 16 different images. We present their results in table (4). In second column, we obtain χ^2 value for cover the image. In third column we calculated χ^2 value for stego image using the JSteg method [3,16]. As we observe a high difference between the second and third columns. As a result, χ^2 -test can recognize the existence of hidden message in stego image. In forth column, χ^2 values was calculated with our new method. Comparison with the second column show the sensible modification and it means that χ^2 values remain nearly fixed and shows deceit of χ^2 algorithm with this method. The existence of message is not revealed in stego image and indeed, this method support the system security against χ^2-test.

References

1. Zhang, T., Ping, X.: A Fast and Effective Steganalytic Technique against JSteg-like Algorithms. In: Proc. 8th ACM Symp. Applied Computing. ACM Press, New York (2003)
2. Judge, J.C.: Steganography: Past, Present, Future, SANS white paper November 30 (2001), http://www.Sans.org/rr/papers/
3. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. IEEE Security and Privacy 1(3), 32–44 (2003)
4. Johnson, N.F., Jajodia, S.: Exploring Steganography: Seeing the Unseen. Computer 31(2), 26–34 (1998)
5. Anderson, R.J., Petitcolas, F.A.P.: On the Limits of Steganography. J. Selected Areas in Comm. 16(4), 474–481 (1998)
6. Liu, Q., Sung, A.H., Ribeiro, B., Wei, M., Chen, Z., Xu, J.: Image complexity and feature mining for steganalysis of least significant bit matching steganography. Information Sciences 178(1), 21–36 (2008)
7. Kurak, C., McHugh, J.: A cautionary note on image downgrading. In: Proceedings of the 8th Computer Security Application Conference, pp. 153–159 (1992)
8. Katzenbeisser, S., Petitcolas, F.A.P.: On Defining Security in Steganographic Systems. In: Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, San Jose, California, vol. 4675 (2002)
9. Cachin, C.: An Information-Theoretic Model for Steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)
10. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG Images: Breaking the F5 Algorithm. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
11. Chen, B., Wornell, G.W.: Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. IEEE Trans. Information Theory 47(4), 1423–1443 (2001)

12. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information Hiding – A Survey. Proceeding of IEEE 87(7), 1062–1078 (1999)
13. Fridrich, J., Goljan, M.: Practical Steganalysis-State of the Art. In: Proc. SPIE Photonics Imaging 2002, Security and Watermarking of Multimedia Contents, vol. 4675, pp. 1–13. SPIE Press (2002)
14. Chen, B., Wornell, G.W.: Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. IEEE Trans. Information Theory 47(4), 1423–1443 (2001)
15. Wallace, G.W.: The JPEG Still Picture Compression Standard. Communications of the ACM 34(4), 30–44 (1991)
16. Westfeld, A., Pfitzmann, A.: Attacks on Steganographic Systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–76. Springer, Heidelberg (2000)
17. Provos, N.: Defending Against Statistical Steganalysis. In: Proc. 10th Usenix Security Symp., pp. 323–335. Usenix Assoc. (2001)
18. Gonzalez, Woods: Digital Image Processing
19. Provos, N., Honeyman, P.: Detecting Steganographic Content on the Internet. In: Proc. 2002 Network and Distributed System Security Symp. Internet Soc. (2002)
20. RSA Data Security. The RC4 Encryption Algorithm (March 1992)
21. Schneier, B.: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In: Anderson, R. (ed.) FSE 1993. LNCS, vol. 809, pp. 191–204. Springer, Heidelberg (1994)
22. Provos, N.: Defending Against Statistical Steganalysis. In: Proceedings of the 10th USENIX Security Symposium, August 2001, pp. 323–335 (2001)
23. Westfeld, A.: F5-A Steganographic Algorithm: High Capacity Despite Better Steganalysis. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)

Near Real Time Enhancement of Remote Sensing Imagery Based on a Network of Systolic Arrays

A. Castillo Atoche[1,2], D. Torres Roman[1], and Y. Shkvarko[1]

[1] Department of Electrical Engineering, CINVESTAV-IPN, 45015 Jalisco, Mexico
[2] Department of Mechatronics, Autonomous University of Yucatan, 150 Yucatan, Mexico
{acastillo,dtorres,shkvarko}@gdl.cinvestav.mx

Abstract. In this paper, we propose a novel Hardware/Software (HW/SW) co-design approach for near real time implementation of high-resolution reconstruction of remote sensing (RS) imagery using an efficient network of systolic arrays (NSA). Such proposed NSA technique is based on a Field Programmable Gate Array (FPGA) and implements the image enhancement/reconstruction tasks of the intelligent descriptive experiment design regularization (DEDR) methodology in an efficient concurrent processing architecture that meets the (near) real time imaging systems requirements in spite of conventional computations. Finally, the results of the HW/SW co-design implementation in a Xilinx Virtex-4 XC4VSX35-10ff668 for the reconstruction of real world RS images are reported and discussed.

Keywords: Remote Sensing, Hardware/Software Co-Design, Network of Systolic Arrays.

1 Introduction

The newer techniques for image reconstruction/enhancement used in high resolution remote-sensing (RS) and radar imaging are computationally expensive [1],[2]. Therefore, these techniques are not suitable for a (near) real time implementation with current digital signal processors (DSP) or personal computers (PC). The descriptive experiment design regularization (DEDR) approach for RS image enhancement/reconstruction has been detailed in many works; here we refer to [3],[4] where such an approach is adapted to the remote sensing (RS) applications with the use of a synthetic aperture radar (SAR) considered in this paper.

The scientific challenge of this study is to solve the enhanced/reconstruction RS imaging problems in context of the (near) real time computing via employing the software/hardware co-design paradigm.

The innovative contribution that distinguishes our approach from the previous studies [4],[5],[6] is twofold. First, we address a new unified intelligent descriptive experiment design regularization (DEDR) methodology and the HW/SW Co-Design technique for (near) real time enhancement/reconstruction of the remote sensing (RS) imagery. Second, the network of Systolic Arrays (NSA) implements the DEDR methodology tasks in a computationally efficient fashion that meets the (near) real time imaging system requirements.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 1070–1077, 2009.

Finally, we report some simulation results and discuss the implementation performance issues related to (near) real time enhancement of the large-scale real-world RS/SAR imagery indicative of the significantly increased performance efficiency gained with the developed HW/SW co-design.

2 Background

In this section, we present a brief summary of the Descriptive Experiment Design Regularization Method (DEDR) previously defined in [3]. Let us consider the measurement data wavefield $u(\mathrm{y})=s(\mathrm{y})+n(\mathrm{y})$ modeled as a superposition of the echo signals s and additive noise n that assumed to be available for observations and recordings within the prescribed time-space observation domain $Y \ni \mathbf{y}$**,** where $\mathrm{y}=(t,\mathrm{p})^{\mathbf{T}}$ defines the time-space points in the observation domain $Y=T\times P$. The model of observation wavefield u is specified by the linear stochastic equation of observation (EO) of operator form [3]: $u=Se+n$; $e\in \mathrm{E}$; $u,n\in \mathrm{U}$; $S:\mathrm{E}\rightarrow\mathrm{U}$. Next, we take into account the stochastic operator-form of the observation equation (EO) vector formwhere

$$\mathbf{u} = \tilde{\mathbf{S}}\mathbf{e} + \mathbf{n} = \mathbf{S}\mathbf{e} + \Delta\mathbf{e} + \mathbf{n}, \tag{1}$$

where the matrix $\tilde{\mathbf{S}} = \mathbf{S} + \Delta_{,}$ represents the nominal signal formation operator SFO and **e**, **n**, **u** represent the zero-mean vectors. These vectors are characterized by the correlation matrices: $\mathbf{R_e} = \mathbf{D} = \mathbf{D}(\mathbf{b}) = \mathrm{diag}\{\mathbf{b}\}$ (a diagonal matrix with vector **b** at its principal diagonal), $\mathbf{R_n}$, and $\mathbf{R_u} = <\tilde{\mathbf{S}}\mathbf{R_e}\tilde{\mathbf{S}}^{+}>_{p(\Delta)} + \mathbf{R_n}$, respectively, where $<\cdot>_{p(\Delta)}$ defines the averaging performed over the randomness of Δ characterized by the *unknown* probability density function $p(\Delta)$.Vector **b** is composed of the elements, $b_k = <e_k e_k^{*}> = <|e_k|^2>$; $k = 1, \ldots, K$, and is referred to as a K-D vector-form approximation of the SSP.

We refer to the estimate, $\hat{\mathbf{b}}$, as a discrete-form representation of the desired SSP i.e. the brightness image of the wavefield sources distributed over the pixel-formatted object scene remotely sensed with an employed array radar/SAR. Thus, one can seek to estimate $\hat{\mathbf{b}} = \{\hat{\mathbf{R}}_{\mathbf{e}}\}_{\mathrm{diag}}$ given the data correlation matrix $\mathbf{R_u}$ pre-estimated by some means, e.g. via averaging the correlations over J independent snapshots [3]

$$\hat{\mathbf{R}}_{\mathbf{u}} = \mathbf{Y} = \underset{j\in J}{\mathrm{aver}}\{\mathbf{u}_{(j)}\mathbf{u}^{+}_{(j)}\} = (1/J)\sum\nolimits_{j=1}^{J}\mathbf{u}_{(j)}\mathbf{u}^{+}_{(j)}\ , \tag{2}$$

and by determining the solution operator that we also refer to as the signal image formation operator (SO) **F** such that

$$\hat{\mathbf{b}} = \{\hat{\mathbf{R}}_{\mathbf{e}}\}_{\mathrm{diag}} = \{\mathbf{F}\mathbf{Y}\mathbf{F}^{+}\}_{\mathrm{diag}}\ . \tag{3}$$

A family of the DEDR-related algorithms for estimating the SSP was derived by [3] as follows.

2.1 Robust Spatial Filtering Algorithm

Consider the white zero-mean noise in observations and no preference to any prior model information, putting $\mathbf{A} = \mathbf{I}$. Let the regularization parameter be adjusted as the inverse of the signal-to-noise ratio (SNR), e.g. $\alpha = N_0/b_0$, where b_0 is the prior average gray level of the SSP, and the uncertainty factor β is attributed to α. In that case the SO **F** is recognized to be the Tikhonov's robust spatial filter:

$$\mathbf{F}_{RSF} = \mathbf{F}^{(1)} = (\mathbf{S}^{+}\mathbf{S} + (N_0/b_0)\mathbf{I})^{-1}\mathbf{S}^{+}. \tag{4}$$

2.2 Robust Adaptive Spatial Filtering Algorithm

Consider the case of an arbitrary zero-mean noise with the composed correlation matrix $\mathbf{R}_{\Sigma}$, equal importance of two error measures, i.e. $\alpha = 1$, and the solution dependent weight matrix $\mathbf{A} = \hat{\mathbf{D}} = \mathrm{diag}(\hat{\mathbf{b}})$. In this case, the SO becomes the robust adaptive (i.e. solution-dependent) spatial filter (RASF) operator:

$$\mathbf{F}_{RASF} = \mathbf{F}^{(2)} = (\mathbf{S}^{+}\mathbf{R}_{\Sigma}^{-1}\mathbf{S} + \hat{\mathbf{D}}^{-1})^{-1}\mathbf{S}^{+}\mathbf{R}_{\Sigma}^{-1}. \tag{5}$$

Now, we are ready to proceed with the algorithms transformation into their locally recursive format representation [10], in which the data dependencies of the computations are exposed in a dependence graphs (DG) [11] to represent the parallel characteristics of the algorithms.

3 Mapping Algorithms onto Systolic Arrays Structures

An array processor consists of a number of processors elements (PE) and interconnection links among the PEs. The systolic design maps an N-dimensional dependence graph DG into a lower dimensional systolic array. In order to derive a regular SA architecture with a minimum possible number of nodes, we employ a linear projection approach for processor assignment, i.e., the nodes of the DG in a certain straight line are projected onto the corresponding PEs in the processor array represented by the corresponding assignment projection vector $\mathbf{d}$. Thus, we seek for a linear order reduction transformation $\mathbf{T}$ [10] where

$$\mathbf{T} = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Sigma} \end{bmatrix}, \tag{6}$$

where $\mathbf{\Pi}$ is a (1 × p)-dimensional vector (composed of the first row of $\mathbf{T}$) which determine the time scheduling and the sub-matrix $\mathbf{\Sigma}$ of $(p - 1) \times p$ dimension (composed of the rest rows of $\mathbf{T}$), determine the space processor [11]. Now, we proceed to construct the SAs structures.

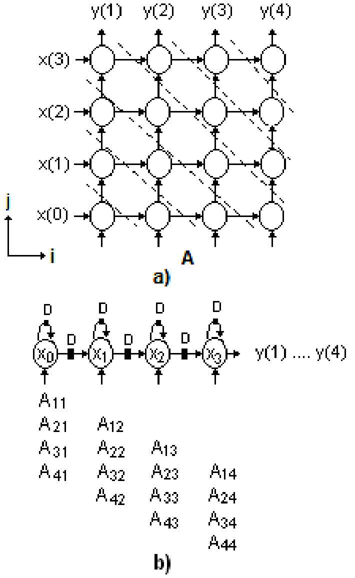

y(1)
y(2)
y(3)
y(4)
x(3)
x(2)
x(1)
x(0)
j
i
A
a)
D
x_0
x_1
x_2
x_3
y(1) .... y(4)
A_{11} A_{21} A_{31} A_{41}
A_{12} A_{22} A_{32} A_{42}
A_{13} A_{23} A_{33} A_{43}
A_{14} A_{24} A_{34} A_{44}
b)

Fig. 1. Matrix-vector systolic implementation: a) Standard DG; b) Systolic array

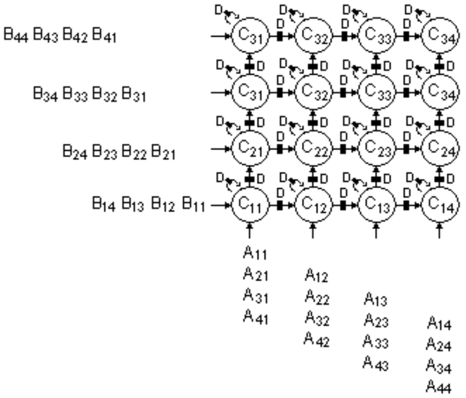

B_{44} B_{43} B_{42} B_{41}
B_{34} B_{33} B_{32} B_{31}
B_{24} B_{23} B_{22} B_{21}
B_{14} B_{13} B_{12} B_{11}
D
C_{31} C_{32} C_{33} C_{34}
C_{31} C_{32} C_{33} C_{34}
C_{21} C_{22} C_{23} C_{24}
C_{11} C_{12} C_{13} C_{14}
A_{11} A_{21} A_{31} A_{41}
A_{12} A_{22} A_{32} A_{42}
A_{13} A_{23} A_{33} A_{43}
A_{14} A_{24} A_{34} A_{44}

Fig. 2. Matrix-matrix systolic implementation

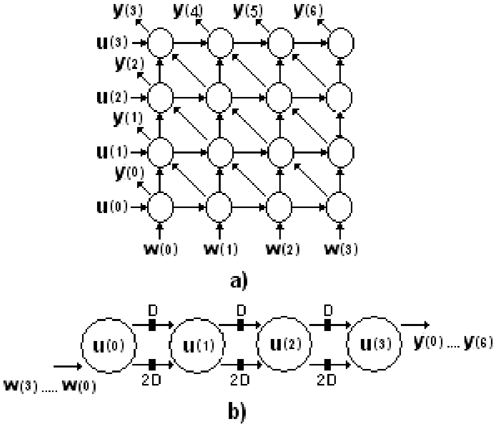


Fig. 3. Convolution systolic implementation: a) Standard DG; b) Systolic array

A. Matrix-Vector Systolic Implementation

Let us consider a matrix **A** of size *n* x *n* and a vector **x** of size *n* x 1, i.e. **y** = **Ax**. The DG for a standard Matrix-Vector multiplication with a vector schedule of $\mathbf{\Pi} = [1\ 1]^T$ is depicted in Figure 2(a). Next, we select a projection vector **d** = $[1\ 0]^{T}$. The corresponding systolic array is obtained as we can see in Figure 2(b). The pipelining period for this systolic array is one. The number of PEs required by this structure is *n.* The computational time required by this systolic array is 2*n*-1 clock periods.

B. Matrix-Matrix Systolic Implementation

Let **A** be an *m* x *n* matrix and **B** be an *n* x *k* matrix. The product of the matrices is an *m* x *k* matrix **C**, i.e. **C=AB**. The DG of a standard matrix-matrix multiplication algorithm corresponds to a 3-D space representation. In Figure 3, the systolic structure with the projection direction of **d**=[0 0 1] ^{T} is obtained. This architecture requires an array of *mk* PEs and *n*+*m*+*k*-1 clock periods.

C. Convolution Systolic Implementation

Given the vectors **u** of size *n* x 1 and **w** of size *m* x 1, the DG of the convolution algorithm with a systolic schedule vector $\mathbf{\Pi}$ =[1 2]^{T} is presented in Figure 4(a). The resulting systolic array considering the projection vector **d**=[1 0] ^{T} is presented in Figure 4(b). This architecture requires an array of *n* PEs and *n*+2*m*-2 clock periods.

4 Hardware/Software Co-design with a Network of Systolic Arrays

The HW/SW co-design is a hybrid method aimed at increasing the flexibility of the implementation and improving the overall design process. The all-software execution of the RS image formation and reconstruction operations may be intensively time consuming, even using modern high-speed personal computers (PC) or any existing digital signal (DSP). In this section, a concise description of a HW/SW co-design approach is presented, and its flexibility in performing an efficient HW implementation of the SW processing tasks with the NSA design is demonstrated. Figure 4 illustrates the pursued HW/SW co-design paradigm. The block units of Figure 4 are to be designed to speed up the digital signal processing operations of the DR algorithm previously developed to meet the real time imaging system requirements.

In this study, we select the Microblaze embedded processor (for the restricted platform) and the On Chip Peripheral Bus (OPB) [7], [8] for transferring the data from/to the embedded processor to/from the NSA as it is illustrated in Figure 4. Such the OPB is a fully synchronous bus that connects other separate 32-bit data buses. Such system architecture (based on the FPGA XC4VSX35-10ff668 with the embedded processor and the OPB buses) restricts the corresponding processing frequency to 100 MHz.

The crucial issue of this design is the proposed NSA design. With the NSA multiple data transfer from the embedded processor data memory to the SAs are avoided. Such NSA design guarantee the drastically reduction of the overall computation time. Finally, the interface unit must employ all the required operational and control functions: loading and storing the data to/from the embedded processor and data/control parallel transfer through the processor elements in a specific spatial/temporal manner. The system control is performed to guarantee the proper synchronization of the data in the proposed interface and to habilitate the corresponding system control of the SA.

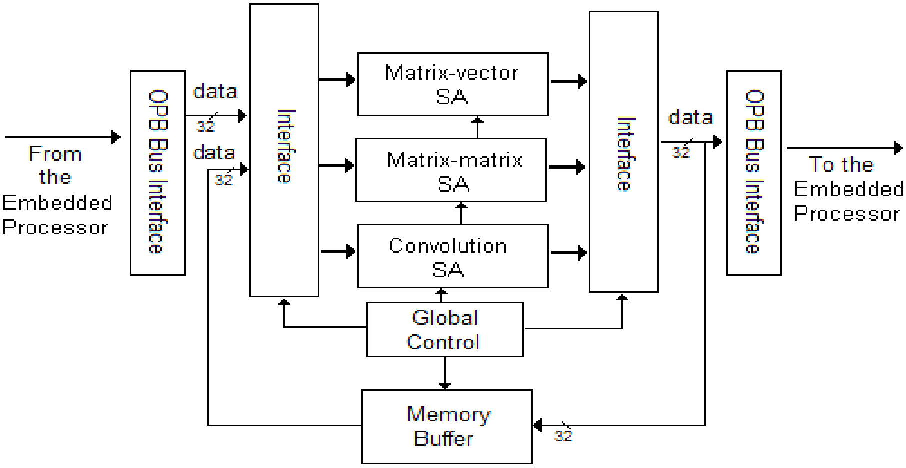


Fig. 4. HW/SW Co-design with the proposed NSA design

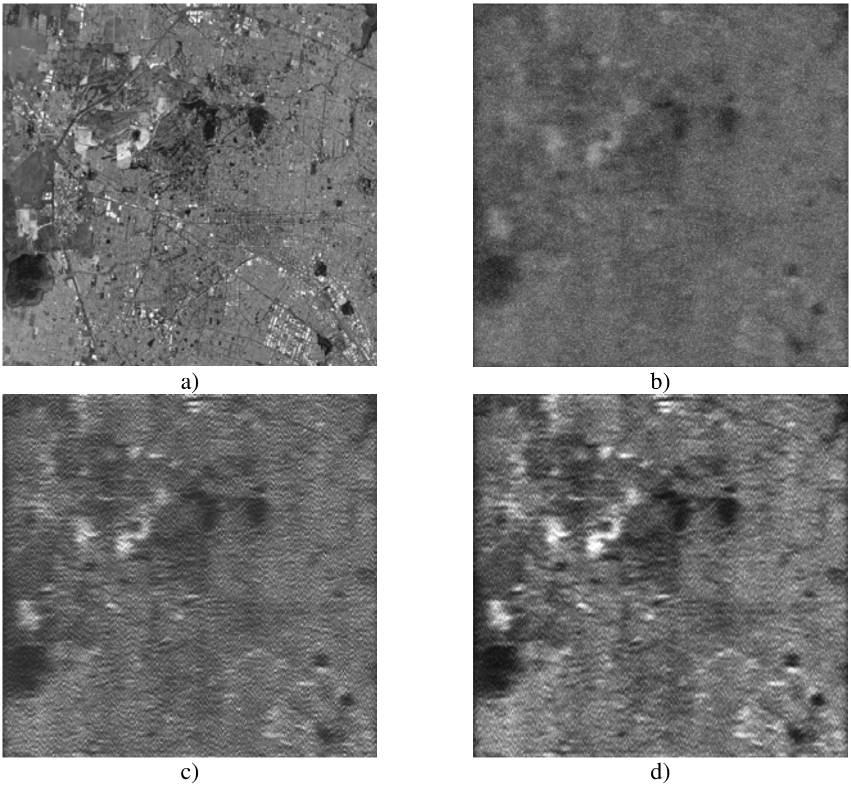


Fig. 5. Operational scenario, scene (μ = 15 dB): (a) original scene; (b) degraded uncertain scene image formed applying the MSF method; (c) image reconstructed applying the RSF algorithm; (d) image reconstructed applying the RASF algorithm

5 Simulations and Performance Analysis

In this study, the simulations were performed with a large scale (512-by-512) pixel format image borrowed from the real-world high-resolution terrain SAR imagery (south-west Guadalajara region, Mexico [9]). The quantitative measures of the image enhancement/reconstruction performance gains achieved with the particular employed DEDR-RSF and DEDR-RASF techniques, evaluated via *IOSNR* metric [3],[6], are reported in Table 1 and Figure 5.

Table 1. Image enhancement of DEDR-related RSF/RASF algorithms

SNR [dB]	**DEDR-regularized RSF Method**	**DEDR-regularized RASF Method**
μ	*IOSNR* [dB]	*IOSNR* [dB]
5	4.31	7.19
10	6.15	9.35
15	7.91	11.01
20	9.31	13.12

Table 2. Synthesis metrics

Area of Hardware Cores	
Logic	Utilization*
Slice Registers, Flip Flops and Latches	14%
LUTs, Logic, Shift Reg. and Dual-RAMs	25%
BUFGs	12%
DSP48	35%

*The reference area is Xilinx Virtex-4 XC4VSX35-10ff668.

Next, the overall timing performances achieved with the proposed approach are reported in Table 3.

Table 3. Timing performances

Timing Performance of Hardware Cores	
Maximum Pin delay:	9.15ns
Average connection delay on the 10 worst nets:	9.07 ns
Maximum Frequency	109.28 MHz

Last, it is compared the required processing time of two different implementation techniques as reported in Table 4. In the first case, the general-form DEDR procedure implemented in the conventional MATLAB software in a personal computer (PC) running at 1.73GHz with a Pentium (M) processor and 1GB of RAM memory and in the second case, the same DR-related algorithms were implemented using the proposed FPGA based HW/SW co-design architecture (partitioning the Matlab application in SW and HW functions) employing the Xilinx FPGA XC4VSX35-10ff668.

Table 4. Comparative time processing analysis

Method	Processing Time [secs]	
	RSF	**RASF**
Optimized-Form DEDR (PC implementation)	19.70	20.05
Proposed HW/SW Co-design with NSA	2.42	2.56

6 Concluding Remarks

The principal result of this paper is the unified intelligent descriptive experiment design regularization (DEDR) methodology and the HW/SW Co-Design technique for enhancement/reconstruction of the remote sensing (RS) imagery using a network of Systolic Arrays (NSA) in a computationally efficient fashion that meets the (near) real time imaging system requirements. We do believe that pursuing the addressed HW/SW co-design paradigm based on NSAs, one could definitely approach the real time image processing requirements while performing the reconstruction of the large-scale real-world RS imagery. Finally, the processing time of the DEDR RSF/RASF algorithms were significantly reduced up to eight times compared with the computational time of the Optimized-form DEDR procedure implemented in the conventional MATLAB software.

References

1. De Micheli, G., Gupta, R.K.: Hardware-Software Codesign. Proceedings of the IEEE V85(3) (1997)
2. Greco, J., Cieslewski, G., Jacobs, A., Troxel, I.A., George, A.D.: Hardware/software interface for high-performance space computing with FPGA coprocessors. In: Aerospace Conference, March 4–11, pp. 10–25. IEEE, Los Alamitos (2006)
3. Shkvarko, Y.V.: Unifying Regularization and Bayesian Estimation Methods for Enhanced Imaging with Remotely Sensed Data. Part I and Part II. In: IEEE Transactions on Geoscience and Remote Sensing, vol. 42, pp. 923–931. IEEE, Los Alamitos (2004)
4. Shkvarko, Y.V., Perez Meana, H.M., Castillo Atoche, A.: Enhanced radar imaging in uncertain environment: A descriptive experiment design regularization paradigm –Accepted for publication. Intern. Journal of Navigation and Observation (2008) (in press)
5. Ponomaryov, V.I.: Real-time 2D–3D filtering using order statistics based algorithms. Journal Real-Time Image Processing 1, 173–194 (2007)
6. Castillo Atoche, A., Shkvarko, Y.V., Perez Meana, H.M., Torres Roman, D.: Convex Regularization-Based Hardware/Software Co-Design for Real-Time Enhancement of Remote Sensing Imagery. Int. Journal of Real Time Image Processing (in press)
7. Meyer Baese, U.: Digital Signal Processing with Field Programmable Gate Array. Springer, Berlín (2001)
8. EDK 9.1 MicroBlaze tutorial in Virtex-4, Xilinx, `http://www.xilinx.com`
9. Space imaging in GeoEye Inc. (2008), `http://www.spaceimaging.com/quicklook`
10. Lo, S.C., Jean, S.N.: Mapping Algorithms to VLSI Array Processors. In: International Conference on Acoustics, Speech, and Signal Processing (1988)
11. Kung, S.Y.: VLSI Array Processors. Prentice Hall, Englewood Cliffs (1988)

Author Index