



Published in final edited form as:

IEEE Trans Med Imaging. 2013 March ; 32(3): 511–530. doi:10.1109/TMI.2012.2224879.

Cortical Surface Reconstruction via Unified Reeb Analysis of Geometric and Topological Outliers in Magnetic Resonance Images

Yonggang Shi,

Laboratory of Neuro Imaging, Department of Neurology, UCLA School of Medicine, Los Angeles, CA 90095, USA

Rongjie Lai,

Department of Mathematics, University of Southern California, Los Angeles, CA 90089, USA

Arthur W. Toga, and

Laboratory of Neuro Imaging, Department of Neurology, UCLA School of Medicine, Los Angeles, CA 90095, USA

Initiative the Alzheimer's Disease Neuroimaging**

Yonggang Shi: yshi@loni.ucla.edu; Rongjie Lai: rongjiei@usc.edu; Arthur W. Toga: toga@loni.ucla.edu

Abstract

In this paper we present a novel system for the automated reconstruction of cortical surfaces from T1-weighted magnetic resonance images. At the core of our system is a unified Reeb analysis framework for the detection and removal of geometric and topological outliers on tissue boundaries. Using intrinsic Reeb analysis, our system can pinpoint the location of spurious branches and topological outliers, and correct them with localized filtering using information from both image intensity distributions and geometric regularity. In this system, we have also developed enhanced tissue classification with Hessian features for improved robustness to image inhomogeneity, and adaptive interpolation to achieve sub-voxel accuracy in reconstructed surfaces. By integrating these novel developments, we have a system that can automatically reconstruct cortical surfaces with improved quality and dramatically reduced computational cost as compared with the popular FreeSurfer software. In our experiments, we demonstrate on 40 simulated MR images and the MR images of 200 subjects from two databases: the Alzheimer's Disease Neuroimaging Initiative (ADNI) and International Consortium of Brain Mapping (ICBM), the robustness of our method in large scale studies. In comparisons with FreeSurfer, we show that our system is able to generate surfaces that better represent cortical anatomy and produce thickness features with higher statistical power in population studies.

Index Terms

Cortical surface reconstruction; Reeb graph; topology correction; Laplace-Beltrami eigenfunctions; tissue classification

Copyright (c) 2010 IEEE.

Correspondence to: Arthur W. Toga, toga@loni.ucla.edu.

**Part of the data used in preparation of this article were obtained from the Alzheimers Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.ucla.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

I. Introduction

Cortical surface reconstruction from magnetic resonance (MR) images is a critical problem in brain mapping which provides the geometric foundation for measuring cortical morphometry and tissue integrity. While many sophisticated algorithms were developed for its solution [1]–[11], significant challenges remain in improving the accuracy, robustness, and speed of cortical reconstruction. In this work, we develop a novel system for the automated reconstruction of cortical surfaces from T1-weighted MR images based on intrinsic analysis of geometry and topology using the Reeb graph of Laplace-Beltrami eigenfunctions. We demonstrate that our system can robustly reconstruct high quality cortical surfaces on large scale data sets.

The successful reconstruction of cortical surfaces requires the development of a complicated image analysis system that includes preprocessing steps such as inhomogeneity correction [12], nonlinear registration [13]–[16], skull stripping [17]–[19], and tissue classification [20]–[23]. Even though different combinations of preprocessing methods were made in various systems, a popular choice for surface formation is that a smooth white matter (WM) surface, which represents the boundary between WM and gray matter (GM), with correct topology is first reconstructed, and then deformed to find the GM surface that represents the boundary of the GM and cerebrospinal fluid (CSF). Due to limited image resolution, partial failure of preprocessing steps, or anatomical variability across population, however, *geometric* outliers, such as spurious spikes, and *topological* outliers, such as handles and tunnels, occur frequently during the surface reconstruction process. To build high quality surface representations of tissue boundaries, the challenge is the removal of both types of outliers without sacrificing accuracy.

Geometric and topological outliers were handled separately in previous works. To avoid geometric outliers in the reconstructed surface, smoothness regularization was applied *globally* together with data terms in surface evolution [4], [5], [7], [8], [11]. This regularization-based method, however, has two problems. First, it only helps but does not guarantee the occurrence of geometric outliers can be prevented, which could be due to leakage into non-cortical areas not removed by skull-stripping. Second, shrinkage can arise, especially in deep sulcal regions, because the regularization is enforced everywhere on the surface, even at places it is unnecessary. This could affect the accuracy in cortical thickness measurements and lead to decreased power in statistical analysis.

To ensure genus-zero topology in reconstructed surfaces, topology-preserving deformations can be used [1], [3], [4], [24], [25] after topological outliers are removed. There are mainly two different approaches for topological outlier detection and correction. The first approach works in the voxel space and uses graph analysis and morphological operations for topology analysis [22], [26], [27], however cut decisions are made according to the geometric size of outlier branches only. The second approach uses the triangular mesh representation of the surface and detects topological outliers from overlapping triangles after mapping the surface to the sphere [28], [29]. Cut/fill decisions are then made according to image intensity and curvature distributions. The drawback of this approach is that it relies on the spherical mapping for outlier detection, which is computationally expensive and not suitable for complicated surfaces with large number of handles and tunnels.

In this work, we present a novel system for automated reconstruction of cortical surfaces from MR images. At the core of our system is a unified approach for the correction of geometric and topological outliers based on intrinsic geometry. With well-composed boundary deformation driven by evolution speeds derived from the MR image [30], [31], our system builds manifold representations of tissue boundaries and computes the Laplace-

Beltrami (LB) eigenfunctions [32]–[43] as Morse functions for Reeb analysis [44]–[49], where we develop a novel algorithm for the efficient construction of Reeb graphs to capture manifold geometry and topology. By analyzing the loops and branches of the intrinsically defined Reeb graph, we develop a unified approach for the detection and correction of geometric and topological outliers. For geometric outliers, intrinsic Reeb analysis naturally leads us to perform localized outlier filtering without causing unintended shrinkage to other parts of the cortex. By computing paired boundary estimates with different topology constraints, our system carefully takes into account information from tissue analysis and geometric regularity for the cut or fill decisions in the correction of topological outliers.

Besides the novel method for geometric and topological outlier correction, there are two other key components in our system. The first is the enhancement of tissue classification with geometric features to help overcome ambiguities in tissue classification. By using features derived from the Hessian of images [50], [51], this improvement enables us to rely on information in the data instead of artificial assumptions to separate touching gyri and alleviate the impact of partial volume effects on misclassification. The second key element is an adaptive interpolation algorithm that achieves sub-voxel accuracy for the final surface. With locally estimated tissue property, we position the surface at sub-voxel locations for the accurate representation of cortical anatomy.

By integrating these novel developments, we build a robust and efficient system for automated cortical reconstruction. An overview of our system is shown in Fig. 1. Given a skull-stripped MR image, we perform enhanced tissue classification and nonlinear registration as described in section II to design initial evolution speeds for the WM and GM surface. Using methods developed in section III and IV, geometric and topological outliers in the WM evolution speed are then corrected with our unified approach, which produces an evolution speed for the estimation of a clean WM boundary. To obtain the GM boundary, we deform the WM boundary outward with the GM evolution speed. Geometric outliers in the GM boundary are removed with intrinsic Reeb analysis. Finally adaptive interpolation developed in section V is applied to both the WM and GM boundary to generate the WM and GM cortical surfaces with sub-voxel accuracy.

The rest of the paper is organized as follows. In section II, we present the tissue enhancement algorithm and derive the evolution speeds for WM and GM boundary estimation. The construction of the manifold representation of tissue boundaries and their Reeb graphs are developed in section III. Using paired boundary estimates, we develop in section IV the unified approach for geometric and topological outlier correction with intrinsic Reeb analysis and localized filtering. In section V, the adaptive interpolation algorithm is developed for sub-voxel accuracy. Experimental results are presented in section VI, where we present detailed comparisons with FreeSurfer [5], [52] and demonstrate that our system can achieve better performance on both simulated and real MR images from two databases. Finally conclusions are made in section VII.

II. Enhancing Tissue Classification

Tissue classification typically uses statistical models to map intensities to three tissue types in the brain: gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF) [20]–[23], but inhomogeneity and partial volume effects usually compromise the effectiveness of such models in correctly identifying thin structures. To enhance fractions of CSF between touching gyri, skeletons of weighted distance transforms from WM were proposed to open up CSF regions [7], but the challenge is that the WM map by itself has inherent uncertainty. In this section, we propose to use features directly derived from the data, which is the eigenstructure of the Hessian, to enhance tissue maps and capture thin structures. Instead of just

enhancing the CSF maps, our method applies to thin structures in both WM and CSF maps, which leads to more integral reconstructions of both WM and GM surfaces. Based on the enhanced tissue maps and atlas labels, we then define evolution speeds for boundary estimation.

Let $I : A \rightarrow \mathbb{R}$ denote the skull-stripped MR image, where A is the lattice of voxels that we represent as $A = \{(i, j, k) \in \mathbb{Z}^3 \mid 0 \leq i < I-1, 0 \leq j < J-1, 0 \leq k < K-1\}$. With partial volume models, a tissue classifier typically uses the intensity at each voxel to assign it as either one of three brain tissue types: GM, WM, CSF, or a weighted combination of WM and GM at the WM boundary, and GM and CSF at the GM boundary. Let p_{GM}, p_{WM}, p_{CSF} denote the individual tissue maps of GM, WM, and CSF, which we compute in this work with the FAST tool in the FSL software [21]. Tissue classification is a critical step in our system as it provides the basis for our enhancement algorithm. We designed our system with the flexibility of taking inputs from any software that generates tissue maps with partial volume models. Here we choose the FAST tool because of its robustness across various datasets. It is also relatively efficient and can compute results in less than 10 minutes. At each voxel, the individual tissue map represents the fraction of the voxel belonging to the corresponding tissue. Voxels where all tissue maps are zero are labeled as background (BG). Using the tissue maps, we can define a composite tissue map as

$$\Phi(\mathbf{p}) = \begin{cases} 0 & \text{BG} \\ p_{CSF}(\mathbf{p}) & \text{if } p_{GM}(\mathbf{p})=0, p_{WM}(\mathbf{p})=0 \\ 1+p_{GM}(\mathbf{p}) & \text{if } p_{CSF}(\mathbf{p})>0, p_{GM}(\mathbf{p})>0 \\ 2 & \text{if } p_{GM}(\mathbf{p})=1 \\ 2+p_{WM}(\mathbf{p}) & \text{if } p_{GM}(\mathbf{p})>0, p_{WM}(\mathbf{p})>0 \\ 3 & \text{if } p_{WM}(\mathbf{p})=1. \end{cases} \quad (1)$$

for all $\mathbf{p} = (i, j, k) \in A$. This composite map provides a simple way of representing the partial volume tissue models. As an example, the composite tissue map of an MR image in Fig. 2 (a), which is from a patient with Alzheimer's disease, is plotted in Fig. 2 (e).

Because tissue classification relies on intensity information to assign tissue types, it does not necessarily follow the geometric assumptions of partial volume models. For example, voxels with small fractions of WM tissue should be either on the boundary of definite WM voxels or *ridges* of the image. When neighboring gyri touch each other, voxels in between with tiny fractions of CSF may exhibit as *valleys* in image intensities. Even though such ridges and valleys have distinctive geometric characteristics, partial volume modeling with intensity distributions may not be sensitive enough to detect them. To overcome this difficulty, we will compute an enhanced tissue map based on Hessian features of the image, which is well-known for ridge detection in image analysis [50], [51].

At a voxel $\mathbf{p} = (i, j, k)$ in the MR image I , we denote its gradient as $\nabla I(\mathbf{p})$, and Hessian as:

$$\text{Hessian}(\mathbf{p}) = \begin{bmatrix} D_{xx}(\mathbf{p}) & D_{xy}(\mathbf{p}) & D_{xz}(\mathbf{p}) \\ D_{yx}(\mathbf{p}) & D_{yy}(\mathbf{p}) & D_{yz}(\mathbf{p}) \\ D_{zx}(\mathbf{p}) & D_{zy}(\mathbf{p}) & D_{zz}(\mathbf{p}) \end{bmatrix} \quad (2)$$

where the entries are the second order gradients of I at this voxel. The eigenvalue of $\text{Hessian}(\mathbf{p})$ with the maximal magnitude is denoted as $\lambda_{max}(\mathbf{p})$, and the corresponding eigenvector is denoted as $v_{max}(\mathbf{p})$. At each voxel, we define the tissue enhancement feature as

$$TE(\mathbf{p}) = \frac{\lambda_{max}(\mathbf{p})}{|\langle \nabla \Psi(\mathbf{p}), v_{max}(\mathbf{p}) \rangle|} \quad (3)$$

where the inner product between $\nabla \Psi(\mathbf{p})$ and $v_{max}(\mathbf{p})$ is the gradient of Ψ along $v_{max}(\mathbf{p})$ at the voxel. For voxels at the top of ridges or bottom of valleys, we have $|\langle \nabla \Psi(\mathbf{p}), v_{max}(\mathbf{p}) \rangle| \approx 0$. Using the sign of $\langle \nabla \Psi(\mathbf{p}), v_{max}(\mathbf{p}) \rangle$, we can determine whether it is a ridge or valley. With this tissue enhancement feature, we can help resolve ambiguities in intensity-based tissue classification. As an example, we show in Fig. 2(b) the tissue enhancement feature. Compared with the image in Fig. 2(a) and the original tissue map in Fig. 2(e), we can see that this feature successfully detects ridges and valleys, but there are also false positives that need to be cleaned.

To filter out these false positives, we use a nonlinear registration software ANTS [16] to register the image to the LPBA40 atlas [53] and warp the probabilistic GM atlas, which assigns to each voxel a probability of belonging to the GM, and anatomical labels to the image space. With the help of nonlinear registration, we can take advantage of atlas information to improve the robustness of our tissue enhancement algorithm and guide the design of evolution speeds for surface reconstruction. Here we choose the ANTS tool for nonlinear warping because of its efficiency in obtaining high quality MR image registration [54]. Typically we can obtain very accurate warped labels and GM atlas in around 30 minutes with the ANTS software. Let p_{GM}^{Atlas} denote the warped GM atlas in the image space.

As an illustration, the GM atlas p_{GM}^{Atlas} is shown in Fig. 2(c), and the warped anatomical labels are shown in Fig. 2(d). Using the anatomical labels, we denote $R_{SUB} = \{\mathbf{p} \mid \mathbf{p} \in \text{putamen, caudate, ventricle}\}$ as the set of voxels belonging to the sub-cortical structures putamen, caudate and ventricles. Using the tissue enhancement feature and atlas labels, we can filter out outliers and detect CSF valleys and WM ridges.

Let $R_{CSF} = \{\mathbf{p} \mid \mathbf{p} \in \text{CSF}, TE(\mathbf{p}) > \tau_{CSF}, p_{GM}^{Atlas}(\mathbf{p}) > 0.05, \mathbf{p} \notin R_{SUB}\}$ be the set of candidate voxels on the CSF valley, where τ_{CSF} is a threshold we typically choose as 2. Because the CSF between touching gyri will occupy tiny fractions of voxels, we apply a thinning algorithm as listed in Table I to R_{CSF} using $-TE$ as the ridge/valley detection feature. Starting from the boundary voxel of R_{CSF} with the least feature value TE , this thinning algorithm iteratively removes voxels if its feature TE is above the threshold τ_{CSF} and is a simple point [24], [55], i.e., peeling it off will not result in a topological change. As a result, we obtain a thin set of voxels R_{CSF} with the maximal tissue enhancing feature.

For partial WM voxels, we detect voxels on the top of ridges to follow the geometry of partial volume models. Let $R_{WM} = \{\mathbf{p} \mid \mathbf{p} \in \text{WM}, 2 < \mathbf{p} < 3, TE(\mathbf{p}) < \tau_{WM}, \mathbf{p} \notin R_{SUB}\}$ be the set of candidate voxels on the ridge, where τ_{WM} is a threshold for ridge detection that we typically choose as -2 . Applying the thinning algorithm to the set R_{WM} with TE as the ridge/valley detection feature, we obtain a thin set of voxels R_{WM} with partial WM.

Using these results, we can define an enhanced tissue map as

$$\hat{\Phi}(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p} \in \hat{R}_{CSF} \\ 3 & \text{if } \mathbf{p} \in \hat{R}_{WM} \\ \Phi(\mathbf{p}) & \text{else.} \end{cases} \quad (4)$$

For the MR image shown in Fig. 2(a), the enhanced tissue map is shown in Fig. 2(f). Compared with the original tissue map in Fig. 2(e), we can see that it successfully opens up

deep sulci in buried gyral regions, and enhances thin WM regions such as the top of the superior frontal gyrus.

Using the enhanced tissue map, we can design evolution speeds for surface generation based on the fast evolution algorithm [31], [56], [57]. By dividing the image into left hemisphere (LH) and right hemisphere (RH) using the warped anatomical labels as shown in Fig. 2(d), we reconstruct the surfaces of each hemisphere separately. In each hemisphere, the evolution speed defined over the grid A for the WM boundary is:

$$F_{WM}(\mathbf{p}) = \begin{cases} 1 & \text{if } \hat{\Phi}(\mathbf{p}) > 2.5 \text{ or } \mathbf{p} \in R_{SUB} \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

and the GM boundary is:

$$F_{GM}(\mathbf{p}) = \begin{cases} 1 & \text{if } \hat{\Phi}(\mathbf{p}) > 1.5 \text{ or } \mathbf{p} \in R_{SUB} \\ -1 & \text{otherwise} \end{cases} \quad (6)$$

The thresholds in (5) and (6) are chosen for the robustness to remaining inhomogeneities and minimization of mesh distortion in reconstructed surfaces. While the correction of inhomogeneity is applied before tissue classification, some remaining inhomogeneities can still exist in various regions of the brain. For example, in regions with hyper-intensities, large blocks of CSF voxels could be included to the interior of the GM boundary if the threshold is selected toward one in (6). On the other hand, in regions with relatively low intensities, large areas of GM voxels could be left out if the threshold is chosen toward two in (6). To achieve a balance in both situations, we choose the thresholds in (5) and (6) such that only voxels containing at least 50% of the WM or GM tissue are included in the WM or GM speed. As illustrated in Fig. 2 (g) and (h), the final reconstructed boundaries are optimally determined with the adaptive interpolation method in section V, which shifts the position of voxel boundaries with locally estimated tissue properties. By putting voxels with more than 50% of the WM or GM tissue to the interior of the boundary and voxels with less than 50% of the WM or GM tissue to the exterior of the boundary, we restrict the movement of all boundary points to be around half of the voxel resolution. This avoids large distortion of triangles and leads to improved mesh quality in reconstructed WM and GM surfaces.

In the next two sections, we will develop the unified approach for the removal of both geometric and topological outliers in the evolution speeds. After that, sub-voxel accuracy will be achieved with the adaptive interpolation scheme in section V to generate the final cortical surfaces.

III. Intrinsic Reeb Graphs on Tissue Boundaries

In this section, we develop numerical algorithms for the construction of intrinsic Reeb graphs on tissue boundaries that correspond to the interface of different tissue types. We first build a mesh representation of tissue boundaries using the evolution speeds computed in section II. With the LB eigenfunction as the Morse function, a novel numerical algorithm is then developed to construct Reeb graphs that intrinsically capture the geometric and topological structure of the tissue boundary.

A. Mesh representation

Starting from an initial mask, we run a fast boundary evolution algorithm, which we developed in our previous work [31], [56], [57], following the evolution speed in (5) or (6). The boundary evolution process separates the lattice of voxels A into two regions: the object

region A_o and the background region A_b . To extract the continuous representation of the boundary between regions, we consider each grid point $\mathbf{p} = (i, j, k)$ as the center point of a rectangular cuboid $\mathcal{C}(\mathbf{p}) = \{(x, y, z) \in \mathbb{R}^3 \mid |x-ih_x| \leq h_x/2, |y-jh_y| \leq h_y/2, |z-kh_z| \leq h_z/2\}$, where h_x, h_y , and h_z are the spatial sampling resolutions in the x, y and z direction of the MR image. Under this cuboid representation, each voxel has six rectangular faces.

Because the well-composedness condition [30] is satisfied in our reconstruction [31], the boundary between A_o and A_b is a manifold and composed of a set of rectangular faces $B = \{B_1, B_2, \dots, B_{N_B}\}$. Each face B_j is the intersection of the cuboids of two voxels: $\mathbf{x}_o^i \in A_o$ and $\mathbf{x}_b^i \in A_b$. This establishes a map from boundary faces to voxels in the object region $\mathbf{x}_o^B: B \rightarrow A_o$ with $\mathbf{x}_o^B(B_i) = \mathbf{x}_o^i$ and the background region $\mathbf{x}_b^B: B \rightarrow A_b$ with $\mathbf{x}_b^B(B_i) = \mathbf{x}_b^i$. By dividing each rectangular face B_j into two triangular faces \mathcal{T}_{2j-1} and \mathcal{T}_{2j} , we have a triangular mesh representation of the boundary $\mathcal{M} = \{\mathcal{V}, \mathcal{T}\}$, where $\mathcal{V} = \{\mathcal{v}_i \mid i = 1, \dots, N_V\}$ and $\mathcal{T} = \{\mathcal{T}_i \mid i = 1, \dots, 2N_B\}$ are the set of vertices and triangles. For each face \mathcal{T}_i , we know it is on the boundary face B_j with $i = \lfloor (j+1)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x . We can define the maps from each triangular face to voxels in A_o and A_b as: $\mathbf{x}_o^T(\mathcal{T}_i) = \mathbf{x}_o^B(B_{\lfloor (i+1)/2 \rfloor})$ and $\mathbf{x}_b^T(\mathcal{T}_i) = \mathbf{x}_b^B(B_{\lfloor (i+1)/2 \rfloor})$. These maps allow us to move freely between the surface representation and voxel representation of the tissue boundary under consideration.

B. Laplace-Beltrami eigen-system

With the triangular mesh representation of the boundary $\mathcal{M} = \{\mathcal{V}, \mathcal{T}\}$, we can compute its LB eigen-system by solving a matrix eigenvalue problem [31]–[33]:

$$Qf = \lambda Uf \quad (7)$$

where λ is the eigenvalue, $f: \mathcal{V} \rightarrow \mathbb{R}$ is the eigenfunction, and the two matrices Q and U are formed using the finite element method [58]. More specifically, the matrices are defined as:

$$Q_{ij} = \begin{cases} \frac{1}{2} \sum_{\mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i)} \sum_{\mathcal{T}_l \in \mathcal{N}(\mathcal{V}_i, \mathcal{V}_j)} \cot \theta_l^{i,j}, & \text{if } i=j; \\ -\frac{1}{2} \sum_{\mathcal{T}_l \in \mathcal{N}(\mathcal{V}_i, \mathcal{V}_j)} \cot \theta_l^{i,j}, & \text{if } \mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i); \\ 0, & \text{otherwise.} \end{cases}$$

$$U_{ij} = \begin{cases} \frac{1}{12} \sum_{\mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i)} \sum_{\mathcal{T}_l \in \mathcal{N}(\mathcal{V}_i, \mathcal{V}_j)} \text{Area}(\mathcal{T}_l), & \text{if } i=j; \\ \frac{1}{12} \sum_{\mathcal{T}_l \in \mathcal{N}(\mathcal{V}_i, \mathcal{V}_j)} \text{Area}(\mathcal{T}_l), & \text{if } \mathcal{V}_j \in \mathcal{N}(\mathcal{V}_i); \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}(\mathcal{V}_i)$ is the set of vertices in the 1-ring neighborhood of \mathcal{V}_i , $\mathcal{N}(\mathcal{V}_i, \mathcal{V}_k)$ is the set of triangles sharing the edge $(\mathcal{V}_i, \mathcal{V}_k)$, $\theta_l^{i,j}$ is the angle in the triangle \mathcal{T}_l opposite to the edge $(\mathcal{V}_i, \mathcal{V}_j)$, and $\text{Area}(\mathcal{T}_l)$ is the area of the l -th triangle \mathcal{T}_l .

The LB eigen-system is discrete and the eigenvalues can be ordered as $0 = \lambda_0 < \lambda_1 < \lambda_2 < \dots$. Correspondingly, the eigen-functions are f_0, f_1, f_2, \dots . The LB eigen-system is intrinsically defined on manifolds and has the nice property of being isometric invariant. It has been applied successfully for various shape analysis tasks in computer vision and medical imaging [32]–[43]. In particular, we will use the first non-constant LB eigenfunction f_1 here as the Morse function for Reeb graph construction [59], which is the solution of the minimization problem

$$\min_{\int_{\mathcal{M}} f d\mathcal{M}=0} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{M} \quad (8)$$

and can be viewed as the smoothest map from the manifold \mathcal{M} to the real line \mathbb{R} . As an illustration, the LB eigenfunction f_1 on a WM boundary surface is plotted in Fig. 5(a). Similar to our previous work in hippocampal modeling with LB eigenfunctions [39], we can see the LB eigenfunction intrinsically models the front-to-posterior trend of the cortical surface.

C. Reeb graph

Given a Morse function f on the mesh \mathcal{M} , its Reeb graph is defined as follows [44].

Definition 1—Let $f: \mathcal{M} \rightarrow \mathbb{R}$. The Reeb graph $R(f)$ of f is the quotient space with its topology defined through the equivalent relation $x \sim y$ if $f(x) = f(y)$ for $x, y \in \mathcal{M}$.

Following this abstract definition, the Reeb graph is intuitively a graph of level contours of f on \mathcal{M} . For the construction of Reeb graphs on surfaces with general topology, previous methods typically need to scan through the whole mesh to detect topological changes of level contours [46]–[48]. Because level contours change topology only at critical points of f , Reeb graph is essentially a graph of critical points. By using level contours at saddle points, an efficient Reeb graph construction algorithm was proposed in [49], but it uses triangle-based region growing and cannot handle densely distributed saddle points. The partition generated by the method in [49] produces non-manifold regions which are not suitable for further analysis with intrinsic geometry. To overcome these drawbacks, we develop a novel method for Reeb graph construction on high genus meshes. The key idea here is that we augment the mesh with level contours crossing the upper and lower neighborhood of saddle points to process close saddle points. In our method, the surface partition generated by the Reeb graph ensures each surface patch is a manifold, so that further analysis of geometry and topology, such as the computation of geodesics, can be performed.

The critical points of f can be classified into maximum, minimum, and saddle points. For a vertex $v_i \in \mathcal{V}$, and its one-ring neighborhood $\mathcal{N}(v_i)$. Let $\mathcal{N}_-(v_i) = \{v_j \in \mathcal{N}(v_i) | f(v_j) < f(v_i)\}$ and $\mathcal{N}_+(v_i) = \{v_j \in \mathcal{N}(v_i) | f(v_j) > f(v_i)\}$ denote the lower and upper neighbors in the 1-ring neighborhood of a vertex v_i . Let $\#$ denote the number of connected components in a set. Using the number of connected components in $\mathcal{N}_-(v_i)$ and $\mathcal{N}_+(v_i)$, we can classify the vertices as follows:

$$CLASS(v_i) = \begin{cases} \max & \#\mathcal{N}_-(v_i)=1, \#\mathcal{N}_+(v_i)=0 \\ \min & \#\mathcal{N}_-(v_i)=0, \#\mathcal{N}_+(v_i)=1 \\ \text{saddle} & \#\mathcal{N}_-(v_i) \geq 2, \#\mathcal{N}_+(v_i) \geq 2 \\ \text{regular} & \text{otherwise.} \end{cases} \quad (9)$$

Let $C = \{C_1, C_2, \dots, C_N\}$ be the set of critical points of f on the surface \mathcal{M} . We assume all critical values are different and sort the critical points according to the critical values such that $f(C_1) < f(C_2) < \dots < f(C_N)$. For anatomical shapes without perfect symmetry, we find this assumption always holds in our experience. For synthetic shapes with perfect symmetry, we can perturb the metric [37], [43] and make sure this assumption is valid. To accurately represent the partition of the surface by neighboring saddle points on the Reeb graph, which could have very subtle differences in the function values, we will augment the original mesh by splitting its triangles along the level contours during the Reeb graph construction process. We denote this augmented mesh as $\bar{\mathcal{M}} = (\bar{\mathcal{T}}, \bar{\mathcal{V}})$, where $\bar{\mathcal{T}}$ and $\bar{\mathcal{V}}$ are the augmented set of

triangles and vertices of M . The Reeb graph is represented as $R(f) = (C, E)$, where $C = \{C_1, C_2, \dots, C_N\}$ is the set of critical points and act as nodes of the graph, and $E = \{E_1, E_2, \dots\}$ is the set of arcs. Each arc in $R(f)$ is represented as $E_i = (SC_i, EC_i, ET_i, LC_i^s, LC_i^e)$, where SC_i, EC_i are the start and end node, ET_i is the set of triangles belonging to this arc in $R(f)$, LC_i^s and LC_i^e are the level contours on the boundary of the arc. The set of triangles ET_i represent the subset of \bar{M} enclosed by LC_i^s and LC_i^e and form a manifold with boundaries.

To construct the Reeb graph, we need to find the arcs connecting these critical points. Let j denote the current arc label that is initialized as -1 . The augmented set of triangles $\bar{\tau}$ and vertices $\bar{\nu}$ are initialized as the original triangle set \mathcal{T} and vertex set \mathcal{V} , respectively. We also assign a label function $ArcLabel$ and initialize it to be -1 for all vertices in $\bar{\nu}$. We scan through these critical points sequentially as follows to build the Reeb graph.

If C_j is a minimum, we increase the current arc label j by one and create a new arc E_j . We set the start node of E_j as $SC_j = C_j$, and its start level contour as $LC_j^s = \{C_j\}$. We also set $ArcLabel(C_j) = j$. Note that this arc is incomplete and we need to find the end node EC_j , the end level contour LC_j^e , and the triangles ET_j .

If C_j is a saddle point, we first define the isovalues of level contours for arcs entering and leaving this node. For the incoming arcs, the isovalue is defined as:

$$f_{iso}^-(C_i) = \frac{f(C_i) + \max(f(C_{i-1}), \max_{\mathcal{V}_j \in \mathcal{N}_-(C_i)} f(\mathcal{V}_j))}{2}. \quad (10)$$

For outgoing arcs, the isovalue is defined as:

$$f_{iso}^+(C_i) = \frac{f(C_i) + \min(f(C_{i+1}), \min_{\mathcal{V}_j \in \mathcal{N}_+(C_i)} f(\mathcal{V}_j))}{2}. \quad (11)$$

The definition ensures that there is no interference from other critical points and that every arc in the Reeb graph would have a manifold structure. This enables us to use existing algorithms for surface analysis such as finding geodesics [60].

For each component in the lower neighborhood $\mathcal{N}_-(C_j)$ of the saddle point C_j , we trace a level contour at the value $f_{iso}^-(C_j)$ on the mesh \mathcal{M} . From the definition of $f_{iso}^-(C_j)$, we know that this level contour crosses all edges between C_j and vertices in this component. This level contour is represented as a polyline $P = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)$ composed of points intersecting edges of \bar{M} at the level $f(\mathbf{p}_i) = f_{iso}^-(C_j)$. We augment the mesh with this set of new vertices by adding edges. Let \mathbf{p}_k and \mathbf{p}_{k+1} be two consecutive points that intersect the two edges of a triangle $\bar{\tau}_{\mathbf{p}_k, \mathbf{p}_{k+1}} = (\bar{\nu}_k, \bar{\nu}_k, \bar{\nu}_k)$ in $\bar{\tau}$. As shown in Fig. 3, we either keep this triangle intact or split it into two or three triangles and add them to $\bar{\tau}$. Repeating this process for all line segment in this level contour, we add all points to $\bar{\nu}$ as new vertices in the mesh. Starting with these new vertices in P , we then grow backward using the algorithm in Table. II and complete an arc E_j in the Reeb graph $R(f)$. We set $EC_j = C_j$, $LC_j^e = P$, and $ET_j = \dots$, which are the set of triangles in $\bar{\tau}$ in the augmented mesh \bar{M} that belong to this arc of the Reeb graph. We repeat this process until all arcs connecting C_j and $\mathcal{N}_-(C_j)$ is crossed by a level contour.

For each component in the upper neighborhood $\mathcal{N}_+(C_i)$ of the saddle point C_i , we trace a level contour at the value $f_{iso}^+(C_i)$ that cross all edges connecting C_i and vertices in this component. Similarly, this level contour is represented as $P = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)$ and all intersecting points in this level contour are added to the augmented vertex set $\bar{\mathcal{V}}$ of the mesh $\bar{\mathcal{M}}$. A new, but incomplete, arc in the Reeb graph is created and we increase the current arc label j by one. We set $SC_j = C_i$, $LC_j^s = P$, and set $ArcLabel(\mathbf{p}_k) = j$ for all newly added vertices $\mathbf{p}_k \in P$. This process is repeated until all edges connecting $\mathcal{N}_+(C_i)$ is crossed.

If C_i is a maximum, we grow backward with it as the starting point using the algorithm in Table. II to complete an arc E_j . We set $EC_j = C_i$, $LC_j^e = \{C_i\}$, and the triangle set $ET_j = \dots$.

By repeating the above process for all critical points, we complete the construction of the Reeb graph $R(f)$. While this Reeb graph is defined on the augmented mesh $\bar{\mathcal{M}}$, which has non-uniform triangles generated from the triangle-splitting process, we can build a map $\mu: \bar{\mathcal{T}} \rightarrow \mathcal{T}$ to relate the properties derived from $R(f)$ to the original mesh \mathcal{M} that has regular triangles. For each triangle $\bar{\tau} \in \bar{\mathcal{T}}$, the map $\mu(\bar{\tau})$ denotes the triangle in \mathcal{T} such that $\bar{\tau}$ is a subset. This is a many-to-one map since the triangles in $\bar{\mathcal{T}}$ is obtained by splitting triangles in \mathcal{T} . By establishing a map between the triangles of \mathcal{M} and $\bar{\mathcal{M}}$, we build a bridge between the Reeb graph on the augmented surface $\bar{\mathcal{M}}$ and the evolution speed defined in the voxel space, which we want to modify with intrinsic Reeb analysis. Once the outliers detected by Reeb analysis are corrected, we can generate the final WM and GM surfaces with the *regular* mesh structure of \mathcal{M} to produce high quality mesh representations.

As an example, we show in Fig. 4 the construction of the Reeb graph on a double torus. The Morse function, which is the first non-constant LB eigen-function of the double torus, is plotted in Fig. 4 (a), where the level contours at a saddle point C_i is illustrated. One level contour is generated that crosses edges connecting the lower neighborhood $\mathcal{N}_-(C_i)$ and this saddle point. Two level contours are generated crossing the edges connecting the upper neighborhood $\mathcal{N}_+(C_i)$ and C_i . The arc construction process is illustrated in Fig. 4 (b), where all arcs are plotted in different colors. For each arc, we also plot its two nodes as black dots. In the first step, an arc between a saddle point and a minimum is constructed. In the 2nd, 3rd, and 4th step, arcs between saddle points are added sequentially to the Reeb graph. In the final step, an arc between a maximum and a saddle point is constructed. The final Reeb graph is visualized in Fig. 4 (c) by plotting triangles on each arc of $R(f)$ with corresponding colors used in Fig. 4 (b), where the graph structure is evident from the neighboring relation of the arcs.

IV. Unified Analysis of Geometric and Topological Outliers

Similar to previous works [5]–[7], our system first reconstructs a clean WM surface with the correct topology and then deform it to obtain the GM boundary. In this section, we develop the unified approach for the removal of geometric and topological outliers on the WM boundary. With topology-preserving evolution, the cleaned WM boundary with genus-zero topology is deformed to generate the GM boundary. While we develop this unified approach in the context of WM surface reconstruction, the method is general and applicable to the analysis of surfaces of arbitrary topology. In particular, we apply it to the GM boundary with spherical topology to remove geometric outliers.

To remove outliers on the WM boundary, we iteratively modify the WM evolution speed with intrinsic Reeb analysis. Using the WM evolution speed, we first compute paired estimates of the WM boundary with and without topology constraints. By comparing the paired boundary estimates, we derive filling voxels for topological artifacts, which provide the basis for cut or fill decisions in topological analysis if the operation is consistent with

underlying tissue properties. By analyzing the intrinsically defined Reeb graph on the WM boundary, we detect geometric and topological outliers and design modifications to the WM evolution speed for their removal. With the modified evolution speed, the above steps are repeated until no change is necessary. Next we describe the details of each step.

A. Paired boundary estimates

Given the WM evolution speed F_{WM} , we use the fast evolution algorithm described in [31] to compute a pair of estimates of the WM boundary for topological analysis. Starting from a genus-zero mask, we evolve the mask toward the boundary under the topology-preserving constraint, which is ensured by only updating points satisfying the simple and well-composedness condition. This generates a genus-zero estimate of the object boundary. The region enclosed by the boundary is denoted as A_o^G and the background region is denoted as A_b^G . The triangular mesh representation of the boundary is $\mathcal{M}^G = (\mathcal{T}^G, \mathcal{V}^G)$. For topological analysis, we turn off the topology-preserving constraint and continue the boundary evolution process under the same speed F_{WM} to obtain the second boundary estimate, which can have arbitrary topology. The object and background region determined by this boundary are denoted as A_o^F and A_b^F , and the boundary mesh is denoted as $\mathcal{M}^F = (\mathcal{T}^F, \mathcal{V}^F)$.

By comparing the paired boundary estimates \mathcal{M}^G and \mathcal{M}^F , we can locate paired patches on \mathcal{M}^G that together fill a handle or tunnel in \mathcal{M}^F . For all faces on \mathcal{M}^G with their interior voxel in the background region of \mathcal{M}^F , i.e., $\mathbf{x}_o^T(\mathcal{F}_i^G) \in A_b^F$, where the map \mathbf{x}_o^T was defined in section III-A, we perform a connected component labeling and denote the set of connected components as $CC^G = \{CC_1^G, CC_2^G, \dots\}$. For faces in the k -th component $CC_k^G \subset \mathcal{F}^G$, we define a connected component (CC) label $L(\mathcal{F}_i^G) = k$. To find paired patches on \mathcal{M}^G that jointly fill a tunnel or handle, we want to find the set of *filling voxels* that connect different components in CC^G and fill topological handles or tunnels. For every voxel \mathbf{p} in $\mathbf{x}_o^T(CC^G)$, which is the set of all interior voxels of faces in CC^G , we define a set $PL(\mathbf{p})$ as the paired labels of connected components it bridges. For each face $\mathcal{F}_i^G \in CC^G$, we first assign its CC label to $PL(\mathbf{x}_o^T(\mathcal{F}_i^G))$. Note that there can be at most two components sharing an interior voxel, so there cannot be more than two labels in the paired label set $PL(\mathbf{p})$ for any voxel \mathbf{p} . Because a subset of the voxels were included in A_o^G to satisfy the well-composedness constraint, they do not directly touch two opposing patches in \mathcal{M}^G . For each $\mathbf{p} \in \mathbf{x}_o^T(CC^G)$ with $\#PL(\mathbf{p}) = 1$, we search its six-neighborhood $\mathcal{N}_6(\mathbf{p})$. If there is a voxel $\mathbf{q} \in \mathcal{N}_6(\mathbf{p})$ that satisfies $\#(PL(\mathbf{q}) \setminus PL(\mathbf{p})) = 2$, which means they together form a bridge to connect two patches, we set $PL(\mathbf{p})$ and $PL(\mathbf{q})$ as the union of these two sets. For each face $\mathcal{F}_i^G \in CC^G$, we assign it the paired label $PL(\mathbf{x}_o^T(\mathcal{F}_i^G))$ of its interior voxel if $\#(PL(\mathbf{x}_o^T(\mathcal{F}_i^G))) = 2$. Using paired labels, we group triangles with the same paired labels into paired components PC_1^G, PC_2^G, \dots and denote $FV_i = \mathbf{x}_o^T(PC_i^G)$ as the set of filling voxels for topological artifacts. As an illustration, we show in Fig. 7 (b) and (d), the filling voxels for a handle and tunnel, respectively. By analyzing the underlying tissue properties on filling voxels, cut or fill decisions could be made in topological analysis with the inclusion of information from tissue classification.

For the unified analysis of geometric and topological outliers on the WM boundary \mathcal{M}^F , we compute its LB eigenfunctions and construct the Reeb graph. Let $f = f_1$ be the first LB eigenfunction of \mathcal{M}^F , as illustrated for a WM boundary in Fig. 5 (a), and $R(f) = (C, E)$ the Reeb graph of f on \mathcal{M}^F , where C denotes the set of critical points, and E the set of arcs. The augmented mesh from the Reeb graph construction is denoted as $\mathcal{M}^F = (\hat{\mathcal{T}}^F, \hat{\mathcal{V}}^F)$. The map from the triangles on $\hat{\mathcal{M}}^F$ to \mathcal{M}^F is denoted as $\mu: \hat{\mathcal{T}}^F \rightarrow \mathcal{T}^F$. Each arc of the Reeb graph

$E_i = (SC_i, EC_i, ET_i, LC_i^s, LC_i^e)$, where $SC_i, EC_i \in C$ are the start and end node, and $ET_i \in \mathcal{T}^F$ is the set of triangles on this arc in $R(f)$, LC_i^s and LC_i^e are the level contours at the start and end node. The degree of a node in the Reeb graph is the number of arcs that it is either the start or the end node. A node is called a leaf node if its degree is one, and an internal node if its degree is greater than one. The arc connected to a leaf node is called a leaf arc.

B. Geometric outlier correction

We detect geometric outliers from the set of leaf arcs in the Reeb graph. Let E_j be a leaf arc with LC_i^s and LC_i^e as the start and end level contour. Using the level contour LC_i^s and LC_i^e , we define an arc feature $H(E_j)$ as follows:

$$H(E_i) = \int_{LC_i^s} \langle \mathbf{p}(s) - \bar{\mathbf{q}}_i^s, \mathbf{n}(\mathbf{p}(s)) \rangle ds + \int_{LC_i^e} \langle \mathbf{p}(s) - \bar{\mathbf{q}}_i^e, \mathbf{n}(\mathbf{p}(s)) \rangle ds \quad (12)$$

where \mathbf{n} is the outward normal on the surface $\bar{\mathcal{M}}^F$, $\bar{\mathbf{q}}_i^s$ and $\bar{\mathbf{q}}_i^e$ are the mean coordinates of points on LC_i^s and LC_i^e . If $H(E_j) > 0$, it means E_j is an outward leaf pointing toward the exterior of the cortex; otherwise, it is an inward leaf that points toward the interior of the cortex.

We consider this leaf arc as an outlier if it satisfies two conditions:

$$\begin{aligned} \text{length}(LC_i^s) + \text{length}(LC_i^e) &< \rho_1 \\ \frac{\text{Area}(ET_i)}{\text{length}(LC_i^s) + \text{length}(LC_i^e)} &> \rho_2 \end{aligned} \quad (13)$$

where $\text{Area}(ET_i)$ is the sum of area of all triangles in ET_i , the parameters ρ_1 and ρ_2 are thresholds selected to identify sharp and small outliers. To further localize geometric outliers, we project the original mesh \mathcal{M}^F onto a subset of its LB eigenfunctions $\mathcal{S} = \{f_0, f_1, f_2, f_3\}$ and calculate the area distortion of triangles. For each triangle in \mathcal{T}^F , its area distortion factor (ADF) is defined as

$$ADF(\mathcal{T}_i^F) = \frac{\text{Area}(\mathcal{T}_i^F)}{\text{Area}(\widehat{\mathcal{T}}_i^F)} \quad (14)$$

where $\text{Area}(\mathcal{T}_i^F)$ and $\text{Area}(\widehat{\mathcal{T}}_i^F)$ are the area of the triangle \mathcal{T}_i^F in the original mesh \mathcal{M}^F and the projected mesh $\bar{\mathcal{M}}^F$. For an outlier leaf arc, we define the set of outlier triangles as $\text{Outlier}(ET_i) = \{ \mathcal{T}_i \mid \mu(ET_i) | ADF(\mathcal{T}_i) > \theta \}$, where $\mu(ET_i)$ is the set of triangles obtained by mapping the triangles in ET_i to the original mesh \mathcal{M}^F , and the parameter θ is a threshold used for further localization of geometric outliers to triangles that exhibit large area distortions during the projection onto the subspace of LB eigenfunctions [31]. To localize outliers on sub-cortical surfaces, the method in [31] relies exclusively on eigen-projection and needs to use hundreds of eigen-functions to form the subspace. For the much more complicated cortical surfaces, we observe in our experiments that a much larger number of eigen-functions are needed to achieve similar localization of outliers. This is computationally very expensive and clearly infeasible for efficient processing because the number of vertices on a cortical surface is typically two orders of magnitude larger than many sub-cortical structures. If we use the same subspace \mathcal{S} of the first four eigen-functions, the eigen-projection method in [31] would produce a large number of false positives, especially on the frontal lobe, as shown in Fig. 5(b). In contrast, we show in Fig. 5(c) and (d) that we can achieve much better performance with the Reeb analysis approach proposed here using the same number of eigen-functions. A zoomed view of detected geometric outliers is plotted in

Fig. 5(c), which shows that the proposed method achieves accurate outlier localization with very few eigen-functions.

To remove the outlier, we modify the evolution speed F_{WM} as follows. If E_i is an outward leaf, we set $F_{WM}(\mathbf{x}_o^T(\mathcal{F}_i^F)) = -1$ for all $\tau_i \in \text{Outlier}(ET_i)$ so that its interior will be removed. If E_i is an inward leaf, we set $F_{WM}(\mathbf{x}_b^T(\mathcal{F}_i^F)) = 1$ for all $\mathcal{F}_i^F \in \text{Outlier}(ET_i)$ such that its exterior will be filled and the outlier can be removed. In contrast to previous methods that rely on global smoothness regularization to avoid geometric outliers, our method only modifies the evolution speed *locally* while leaving other parts of the boundary intact. This greatly decreases shrinkage effect and help improve the accuracy in surface reconstruction.

C. Topological outlier correction

For topological analysis, we first remove duplicated arcs in $R(f)$. For any two arcs with the same start and end node, we remove the one with smaller size from $R(f)$ and add it as a topological outlier to a set we denote as TO . After that, we can represent the Reeb graph $R(f)$ as a matrix W and apply standard graph search algorithms to locate the remaining topological outliers. For any arc $E_i \in R(f)$, we set $W(i_1, i_2) = \text{Area}(ET_i)$, where i_1 and i_2 are the indices of the start and end critical points, respectively, and $\text{Area}(ET_i)$ is the area of this arc.

We detect topological outliers from redundant paths between saddle points in the Reeb graph. Using the matrix representation of the Reeb graph, we analyze paths starting at each node to locate topological outliers. For a node C_j in the directed graph W , if there are more than one out-going arcs, we use breadth-first-search (BFS) to test if an outlier exists. Let j_1, \dots, j_K denote the set of neighboring nodes which satisfies $W(i, j_1) > 0, \dots, W(i, j_K) > 0$. For every outgoing arc, we set $W(i, j_k) = 0$ and perform BFS with C_j as the starting node. This generates a spanning tree of nodes reachable from C_j after the removal of the arc (C_j, C_{j_k}) . The spanning tree is recorded as an array PAR_k . For each node C_j , $PAR_k(C_j)$ is the parent node in this spanning tree if it is reached by C_j from $PAR_k(C_j)$. By repeating this step for all outgoing arcs at C_j , we generate a set of BFS trees PAR_1, \dots, PAR_K . The intersection node $C_{J_{min}}$ of these trees is defined as the node with the smallest index j such that $PAR_k(C_j) > 0$ for all $k = 1, \dots, K$. Starting from the intersection node $C_{J_{min}}$, we trace backward to the current node C_j on each BFS tree PAR_k and record the path as $PATH_k = (C_j, \dots, PAR_k(C_{J_{min}}), C_{J_{min}})$. The cost of this $PATH_k$ is the total area of all arcs on this path. Among all paths we pick the one with the smallest cost and add all arcs on this path to the outlier set TO as topological outliers. As an illustration, the topological outliers detected with Reeb analysis on the surface in Fig. 5(a) are plotted in red in Fig. 5(c) and (d). A zoomed view of a topological outlier is plotted in Fig. 5(d).

To provide further localization of topological operations, we compute cutting paths on the handles and tunnels detected by Reeb analysis. For each arc in the outlier set TO , we first decide if this is a handle or tunnel on the boundary and then apply different analysis strategies to calculate a cutting path. For better illustration, we show in Fig. 6 the two possible cases that a topological outlier can be represented in the Reeb graph. In Fig. 6(a), the arc of the Reeb graph, which is composed of the triangles plotted in red, captures this outlier attached to a box as a handle. On the other hand, this outlier could also be captured by an arc that forms a tunnel as plotted in Fig. 6(b). To differentiate these two cases, we compute the arc feature $H(E_i)$ as defined in (12) for each arc $E_i \in TO$. If $H(E_i) > 0$, which means the centroids of the start and end level contours are enclosed by the surface as in the case plotted in Fig. 6(a), we classify E_i as a handle. If $H(E_i) < 0$, which means the centroids of the start and end level contours fall outside the surface as shown in Fig. 6(b), we classify E_i as a tunnel. For a handle or tunnel, we use a different method to find a cutting path with

minimal length if a cut operation is needed. If E_j is a handle, we uniformly sample a set of level contours of the eigenfunction f between the start contour LC_i^s and end contour LC_i^e , and pick the contour with the least length as the cutting path. For example, a handle on a WM boundary is plotted in red in Fig. 7(a) and the cutting path as the shortest level contour is plotted in Fig. 7(b). For a tunnel, we combine two geodesic paths to form a cutting path. The first geodesic goes from the start to the end node within the triangles ET_j . The second geodesic goes from the start to the end node without passing through the triangles in ET_j . For a tunnel on a WM boundary as shown in Fig. 7(c), the cutting path that is composed of two geodesics is plotted in Fig. 7(d).

We use information from tissue classification and geometric regularity to make cut/fill decisions about topological outliers. Based on tissue maps, we enforce the anatomical knowledge that gyri on WM surfaces should not enclose CSF. Using the area distortion during eigen-projection, we apply the assumption that the WM surfaces should have similar geometric regularity as a folded sheet. From the analysis of the paired boundary estimates \mathcal{M}^G and \mathcal{M}^F , we have paired patches PC_1^G, PC_2^G, \dots on the genus-zero boundary estimate \mathcal{M}^G . These patches fill the handles and tunnels on \mathcal{M}^F . For each paired patches, its interior voxels are $\mathbf{x}_o^T(PC_i^G)$ and they are filled inside A_o^G to satisfy the genus-zero constraint, which are illustrated in Fig. 7 for a handle and tunnel. At a paired patch PC_i^G , a cut decision should be made if either of the following two conditions is met

- If the number of voxels $\mathbf{p} \in \mathbf{x}_o^T(PC_i^G)$ with $\tau(\mathbf{p}) = 1$ is greater than THD_{CSF} .
- If the number of triangles $\mathcal{T}_j \in PC_i^G$ with $ADF(\tau_j) > \tau_{th}$ is greater than THD_{GEO} .

Both parameters THD_{CSF} and THD_{GEO} are thresholds we choose empirically to identify topological outliers that are not consistent with the underlying image intensity distributions and geometric regularity of cortical surfaces. The first condition checks if filling a handle or tunnel needs more than THD_{CSF} voxels classified as CSF in the enhanced tissue map. The second condition measures the geometric regularity of the filling patches by calculating if there are more than THD_{GEO} triangles with its ADF, which is defined in (14), greater than the threshold τ_{th} used for geometric outlier detection.

To implement the cut/fill decision from the analysis of paired patches, we use the handles and tunnels detected by the Reeb analysis process. To cut open a paired component in \mathcal{M}^G , we pick a cutting path on a handle or tunnel that is connected to this paired component. For the cutting path CP_j of an arc E_j in $R(f)$, we denote $\tau^F(CP_j)$ as the set of triangles it passes on \mathcal{M}^F . We consider a paired component PC_i^G connected to a cutting path CP_j if the set $\mathbf{x}_b^T(\mathcal{T}^F(CP_j)) \cap \mathbf{x}_o^T(PC_i^G)$ is not empty, which means the exterior voxels of faces on the cutting path intersects with the interior voxels of the paired patches. Among all cutting paths intersecting the paired component, we choose the one with the shortest length and denote it as $CP_{J_{min}}$. To cut it open, we modify the evolution speed as

$$F_{WM}(\mathbf{p}) = -1 \quad \forall \mathbf{p} \in \mathbf{x}_o^T(\mathcal{T}^F(CP_{J_{min}})). \quad (15)$$

Similar to the modification for the removal of geometric outliers, this change to the evolution speed is *local* and peels off a layer of voxels along the cutting path. To completely cut open a large outlier, this process might be applied to the same outlier during each iteration of our unified outlier correction algorithm. The number of iterations it takes for the topology correction algorithm to converge depends on the complexity of the topological

outliers in the mask boundary. In our experience from experiments on various MR datasets, this process typically takes less than five iterations to converge.

D. WM and GM boundary estimate

With the modified speed obtained from the geometric and topological analysis process in section IV-B and IV-C, we can repeat the above steps in section IV-A, IV-B and IV-C until no changes are made to the WM speed. The genus zero estimate \mathcal{M}^G then gives a clean estimate of the WM boundary. For the example in Fig. 2(a), the cleaned WM boundary after the removal of outliers shown in Fig. 5 is plotted in Fig. 8(a).

In order to reconstruct the GM boundary, we evolve the cleaned WM boundary outward using the GM speed defined in (6) under the topology-preserving constraint. To remove geometric outliers on the GM boundary estimate, the Reeb analysis is applied to this genus zero surface and the geometric outlier correction process described in section IV-B is applied iteratively. By removing the spurious outliers, we obtain a clean estimate of the GM boundary. As an illustration, the cleaned GM boundary corresponding to the example in Fig. 2 is plotted in Fig. 8(c).

V. Sub-voxel Accuracy with Adaptive Interpolation

After removing geometric and topological outliers, we obtain a clean reconstruction of the boundary between two tissue types. For the WM surface, it is the boundary between WM and GM. For the GM surface, it is the boundary between GM and CSF. To achieve sub-voxel accuracy in the final surface reconstruction, we develop an adaptive interpolation method in this section.

For a tissue boundary between a high intensity tissue \mathcal{I}_h and low intensity tissue \mathcal{I}_l , we denote $B = (B_1, B_2, \dots, B_{N_B})$ as the set of rectangular faces and $\mathcal{M} = (\mathcal{V}, \mathcal{T})$ the triangular mesh representation of the boundary surface as defined in section III-A. The pair (i, j) could be $(3, 2)$ for the WM boundary or $(2, 1)$ for the GM boundary. For a face B_i , its interior voxel neighbor is $\mathbf{x}_o^B(B_i)$ and exterior voxel neighbor is $\mathbf{x}_b^B(B_i)$. We denote BV_{in} and BV_{out} as the set of interior and exterior neighboring voxels of all rectangular faces on the boundary, respectively.

For each voxel $\mathbf{p} \in BV_{in} \cup BV_{out}$, its cuboid $\mathcal{C}(\mathbf{p})$ has two faces in each of the x , y , and z direction. To calculate the shift of faces for partial volume modeling, we count the number of faces that the intersection of $\mathcal{C}(\mathbf{p})$ and B , i.e., $\mathcal{C}(\mathbf{p}) \cap B$, have in the x , y , z direction and denote them as $N_x(\mathbf{p})$, $N_y(\mathbf{p})$, and $N_z(\mathbf{p})$, respectively. We also estimate locally the image intensity $\Psi_h(\mathbf{p})$ for the high intensity tissue \mathcal{I}_h as the mean intensity of voxels classified as \mathcal{I}_h in the $5 \times 5 \times 5$ neighborhood centered at \mathbf{p} . Similarly, the image intensity $\Psi_l(\mathbf{p})$ of the low intensity tissue \mathcal{I}_l is estimated locally as the mean intensity of voxels classified as \mathcal{I}_l in the same neighborhood. The fraction of high intensity tissue \mathcal{I}_h contained in this voxel \mathbf{p} can then be computed with linear interpolation as:

$$\alpha(\mathbf{p}) = \begin{cases} 1 & \text{if } \Psi(\mathbf{p}) \geq \Psi_h(\mathbf{p}); \\ 0 & \text{if } \Psi(\mathbf{p}) \leq \Psi_l(\mathbf{p}); \\ \frac{\Psi(\mathbf{p}) - \Psi_l(\mathbf{p})}{\Psi_h(\mathbf{p}) - \Psi_l(\mathbf{p})} & \text{otherwise.} \end{cases} \quad (16)$$

To account for this partial volume effect, we will shift faces on the boundary. Let $\mathcal{D}(\mathbf{p}) = \min(1, N_x(\mathbf{p})) + \min(1, N_y(\mathbf{p})) + \min(1, N_z(\mathbf{p}))$ be the total number of directions that we will shift faces. The shrink factor in each direction is then $\alpha(\mathbf{p})^{1/\mathcal{D}(\mathbf{p})}$.

For a face B_i , if the tissue map of its interior voxel $\Phi(\mathbf{x}_o^B(B_i)) < \Phi_{h_i}$, it needs to move inward according to the shift factor $(\mathbf{p}^*) = 1 - (\mathbf{p}^*)^{1/(\mathbf{p}^*)}$ computed with $\mathbf{p}^* = \mathbf{x}_o^B(B_i)$. If $\Phi(\mathbf{x}_o^B(B_i)) = \Phi_{h_i}$, it needs to move outward according to the partial volume shift factor $(\mathbf{p}^*) = (\mathbf{p}^*)^{1/(\mathbf{p}^*)}$ computed from $\mathbf{p}^* = \mathbf{x}_o^B(B_i)$. Let $\mathbf{p}^* = (p_x^*, p_y^*, p_z^*)$ denote the coordinate of the point \mathbf{p}^* . We also denote $\hat{\mathbf{p}}^* = (\hat{p}_x^*, \hat{p}_y^*, \hat{p}_z^*)$ as the projection of \mathbf{p}^* onto the face B_i , which is the nearest point on the face B_i . The shift for this face is

$$S_B(i) = \begin{cases} \left(-\frac{\alpha'(\mathbf{p}^*)}{N_x(\mathbf{p}^*)} h_x, 0, 0 \right) & \text{if } \mathbf{p}_x^* - \hat{p}_x^* < 0 \\ \left(\frac{\alpha(\mathbf{p}^*)}{N_x(\mathbf{p}^*)} h_x, 0, 0 \right) & \text{if } \mathbf{p}_x^* - \hat{p}_x^* > 0 \\ \left(0, -\frac{\alpha'(\mathbf{p}^*)}{N_y(\mathbf{p}^*)} h_y, 0 \right) & \text{if } \mathbf{p}_y^* - \hat{p}_y^* < 0 \\ \left(0, \frac{\alpha(\mathbf{p}^*)}{N_y(\mathbf{p}^*)} h_y, 0 \right) & \text{if } \mathbf{p}_y^* - \hat{p}_y^* > 0 \\ \left(0, 0, -\frac{\alpha'(\mathbf{p}^*)}{N_z(\mathbf{p}^*)} h_z \right) & \text{if } \mathbf{p}_z^* - \hat{p}_z^* < 0 \\ \left(0, 0, \frac{\alpha(\mathbf{p}^*)}{N_z(\mathbf{p}^*)} h_z \right) & \text{if } \mathbf{p}_z^* - \hat{p}_z^* > 0 \end{cases} \quad (17)$$

where the shift in each direction is multiplied by the corresponding spatial resolution, and divided by the number of faces that $\mathcal{C}(\mathbf{p}^*) = B$ has in that direction.

Given the shift of all faces in B , we calculate the shift for a vertex \mathcal{V}_i as the average shift of its neighboring faces:

$$S_{\mathcal{V}_i}(i) = \frac{\sum_{B_j \in \mathcal{N}_R(\mathcal{V}_i)} S_B(j)}{\#\mathcal{N}_R(\mathcal{V}_i)} \quad (18)$$

where $\mathcal{N}_R(\mathcal{V}_i)$ denote the set of rectangular faces in the neighborhood of the vertex \mathcal{V}_i , and $\#\mathcal{N}_R(\mathcal{V}_i)$ is the number of rectangular faces in this set.

Let \mathbf{x} denote the vector of coordinates of all vertices, and $S_{\mathcal{V}}$ the shift of all vertices. We compute the final vertex coordinates with sub-voxel accuracy by minimizing the following energy function:

$$\mathcal{E}(\hat{\mathbf{x}}) = \|\mathbf{x} + S_{\mathcal{V}} - \hat{\mathbf{x}}\|^2 + \eta \|\Delta \hat{\mathbf{x}}\|^2 \quad (19)$$

where \mathbf{x} is the optimized vertex coordinate, Δ is the discrete Laplacian matrix of the mesh, and η is a regularization parameter. Note that we have chosen the Laplacian instead of the gradient operator in the regularization term to avoid large shrinkage effects from the gradient operator. The solution of this quadratic problem gives us the coordinates of vertices on the smoothed surface:

$$\hat{\mathbf{x}} = (I + \eta \Delta' \Delta)^{-1} (\mathbf{x} + S_{\mathcal{V}}). \quad (20)$$

¹Part of the data used in the preparation of this article were obtained from the Alzheimers Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a \$60 million, 5- year public-private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimers disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials.

By applying the sub-voxel calculation to both the WM and GM surfaces, we obtain our final solution for cortical surface reconstruction. As an illustration, we have plotted the surfaces before and after sub-voxel interpolation in Fig. 8. Clearly we can see the improved quality in the reconstructed surfaces after the application of the adaptive interpolation scheme.

VI. Experimental Results

In this section, we present experimental results on 40 simulated MR images and T1-weighted real MR images of 200 subjects to demonstrate that our method can reconstruct high quality cortical surfaces on large scale data sets. We also compare our results with surfaces reconstructed with FreeSurfer [5], [52], a widely used software for cortical surface reconstruction and analysis in brain imaging research, to demonstrate that our method can obtain better results with significantly less computational cost.

The simulated dataset includes a set of 20 simulated MR images from a publicly available dataset [61], and a set of 20 images we simulated with known gray matter atrophy. The real T1-weighted MR images used in our experiments are from two publicly available databases. The first dataset has 100 baseline MR images, from 50 normal controls (NC) and 50 Alzheimer's disease (AD) patients, from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [62]¹. The second dataset includes MR images of 100 subjects from the International Consortium of Brain Mapping (ICBM) [63]. The 100 subjects have a wide age range from 19 to 80.

All MR images were first automatically skull-stripped with a meta algorithm [64] in the LONI pipeline [65]. Using the skull-stripped images, our system and FreeSurfer automatically reconstructed the WM and GM surfaces on the left and right hemispheres in all subjects. For our method, the same set of parameters are used: $r_1 = 100mm$, $r_2 = 5$, $r_3 = 100$, $THD_{CSF} = 5$, $THD_{GEO} = 5$, $\epsilon = 10$. For FreeSurfer, the default settings were used. All computations are carried out on the grid of the LONI pipeline. In general, our system takes around 2–4 hours to reconstruct surfaces from each subject. For FreeSurfer, the surface reconstruction process typically takes around 10 to 20 hours, and the whole workflow, including surface labeling, can take 20 to 30 hours.

A. Qualitative comparisons

As an illustration, we first present in this experiment a detailed comparison between our method and FreeSurfer using the same MR scan plotted in Fig. 2. This is an MR image of an AD patient from the ADNI dataset. Representative examples from both the ADNI and ICBM datasets will then be presented to demonstrate that our method can reconstruct surfaces that better preserve the integrity of cortical anatomy.

For the MR image of an AD patient, we applied our method and FreeSurfer to reconstruct the WM and GM surfaces on both hemispheres. Computationally our method took around 3 hours and is much more efficient than FreeSurfer, which took over 10 hours. Both our method and FreeSurfer successfully reconstructed cortical surfaces with genus-zero topology. The unified outlier removal process of our method took around 20 minutes for each hemisphere. The topological correction process in FreeSurfer took more than one and half hour for each hemisphere. As a more detailed illustration of the Reeb analysis process for outlier removal, we plotted in Fig. 9 the number of nodes in the Reeb graph, the number of geometric outliers, and the number of total handles and tunnels at each iteration of our algorithm. With the increase of iteration, we can see the complexity of the Reeb graph decreases with the number of outliers in the boundary. For this image, it took 3 iterations for the unified outlier correction algorithm to converge when all geometric outliers were removed and no more cuts to be made. After that, a topology-preserving evolution was

applied to reconstruct a genus-zero boundary. The adaptive interpolation scheme developed in section V was finally applied to reconstruct a smooth and accurate representation of the cortical surface. As an illustration, we plotted the intersection of the image slice shown in Fig. 2(a) with the WM and GM surfaces reconstructed by our method in Fig. 2(g) and (h). From the results we can see that our method is able to robustly reconstruct the WM and GM boundaries even in the presence of tissue degeneration such as the white matter degeneration on the right hemisphere of this AD patient.

To compare the results from our method and FreeSurfer, the left hemispherical WM surfaces reconstructed by both methods are plotted in Fig. 10. The two WM surfaces have similar number of vertices because they are both derived from the WM mask boundary. The FreeSurfer WM surface has 137309 vertices, and our WM surface has 138130 vertices. In regions highlighted in the dashed circles in Fig. 10, we can see that our method produces more complete reconstruction of the boundary, which is better illustrated with the intersections of surfaces and two axial slices shown in Fig. 10(c) and (f). The left hemispherical GM surfaces reconstructed with our method and FreeSurfer are plotted in Fig. 11. Because the FreeSurfer GM surface is obtained by deforming the WM mesh to the GM boundary, it has the same number of vertices as its WM surface and typically has irregular triangles. On the contrary, our method follows the boundary between GM and CSF to generate a more uniform mesh representation of the GM surface. The GM surface reconstructed with our method here has 145136 vertices, which has slightly more vertices than the WM surface. This is natural in order for the GM surface to have the same level of vertex density and accuracy as the WM surface because it has a larger area than the WM surface. From the highlighted mesh structures of four ROIs in Fig. 11, we can see that FreeSurfer produced *broken gyri* in these regions while our method is able to generate more complete reconstruction at these gyri. The intersections of the GM surfaces reconstructed by our method and FreeSurfer with three sagittal slices are shown in Fig. 12, which illustrate that our surface can better capture deep sulcal regions. This shows that the localized outlier detection and filtering approach in our method is able to avoid shrinkage and produce a more accurate surface representation in these regions.

To further demonstrate that surfaces generated by our method are able to better represent the integral anatomy of the cortex, we have plotted in Fig. 13 more representative reconstruction results from ADNI and ICBM data. For each subject, the left hemispherical GM surfaces reconstructed by our method and FreeSurfer are plotted with differences highlighted in dashed circles. Compared with results from our method, we can see that FreeSurfer results failed to reconstruct a complete gyrus in the regions enclosed in the dashed circles. This shows that our method is able to produce a more complete representation of cortical anatomy.

B. Atrophy detection on simulated MR images

In this experiment, we will compare our method and FreeSurfer on the detection of simulated atrophy in MR images where the ground truth of WM, GM, and CSF tissue maps are known. By matching reconstructed cortical surfaces with the underlying tissue boundaries, we can analyze the performance of different topology correction strategies in our method and FreeSurfer. Using the reconstructed WM and GM surfaces, we can calculate cortical thickness and compare quantitatively the ability of these two methods in detecting *sub-voxel* tissue atrophy.

To simulate longitudinal brain atrophy, we first downloaded from BrainWeb [66] a set of 20 simulated MR images of normal subjects with known WM, GM, and CSF tissue maps [61] and used them as the baseline images. All images have an isotropic spatial resolution of 1mm. Using these tissue maps, we simulated a 0.1 mm GM atrophy everywhere on the

cortex by modifying intensities in the MR images. For each voxel with more than 10 percent GM tissue and on the boundary of GM and CSF, we introduced a sub-voxel atrophy by subtracting its intensity by 10 percent of the difference between the average GM intensity and CSF intensity in the image because we assumed the lost GM tissue in this voxel will be replaced by CSF. By applying this procedure to all baseline images, we obtained a set of 20 images with simulated GM atrophy, which we denoted as follow-up scans of these 20 subjects. Both our method and FreeSurfer were applied to the 40 simulated images to reconstruct the WM and GM cortical surfaces.

From the examples shown in Fig. 10 and Fig. 13, we see various cases that our method is able to generate a more integral reconstruction of the superior temporal cortex than FreeSurfer. With simulated images, we have the opportunity to demonstrate this on images with known tissue boundaries. In Fig. 14(a) and (d), we plotted the left WM surfaces reconstructed from one of the baseline images with our method and FreeSurfer, where we can clearly see that our result has a more integral representation of the superior temporal cortex. For both methods, we extracted the WM boundary before topology correction to investigate the cause of this difference. In Fig. 14 (b) and (d), we plotted the corresponding WM boundary without topology correction in the region highlighted by the dashed ellipses in Fig. 14 (a) and (d), respectively. In both cases, we can see a hole is clearly present on the superior temporal gyrus. Different correction strategies were adopted by these two methods: our method chose to fill the hole, while FreeSurfer decided to cut it open. As a result, different surface reconstruction results were obtained as shown in Fig. 14(c) and (f). The intersection of the reconstructed surfaces with two sagittal slices of the underlying true WM tissue map were plotted in Fig. 14 (g) and (h), where our result was plotted in red and the FreeSurfer result was plotted in blue. As highlighted by the green arrows in these two pictures, we can see that our result provides a more faithful representation of the underlying tissue boundary at the superior temporal cortex.

To compare the performance of our method and FreeSurfer in the detection of simulated longitudinal atrophy, we parcellated all cortical surfaces into gyral regions and tested for GM thickness changes in each region. In this work, we used the gyral labels on GM surfaces generated by FreeSurfer for a localized comparison. In order to project the FreeSurfer label onto our surface, we aligned both surfaces in the image space. For each vertex on our surface, we found the nearest vertex on the FreeSurfer surface such that the normal directions of both vertices have positive inner products and pulled back the label on this vertex. As an illustration, we plotted in Fig. 15 the labels on the GM surfaces generated by FreeSurfer and our method as shown in Fig. 11. We can see that we have successfully mapped the gyral labels onto the GM surface generated by our method. At each point of a GM surface, we used its distance to the closest point on the WM surface as the measure of thickness [7], [67]. Numerically, we first computed the signed distance function of the WM surface with the fast marching method [68], and then calculated the thickness at each vertex of the GM surface with sub-millimeter accuracy by interpolating the value of the signed distance function. For surfaces generated by both our method and FreeSurfer, the same approach of thickness calculation was applied for consistency. On results generated by both our method and FreeSurfer, we calculated the change of mean thickness on each gyrus between the follow-up and baseline scan of each subject. On each gyrus, a two-tailed t-test was applied to the thickness changes of the 20 subjects and tested if we could reject the null-hypothesis that there is no significant change of thickness between the baseline and follow-up scan. For cortical surfaces generated by our method and FreeSurfer, the p-values obtained from the t-tests at each gyrus were mapped onto the corresponding gyrus of a subject and plotted in Fig. 16, where p-values larger than 0.01 were considered insignificant and plotted in light gray. From the results, it is clear that our method was able to detect significant atrophy in more regions and generate more significant p-values than FreeSurfer. More

specifically, on the left hemisphere, our results produced more significant p-values on 20 of the 34 gyral regions. On the right hemisphere, our results produced more significant p-values on 25 of the 34 gyral regions. Next we will demonstrate that our method can achieve more statistically significant results than FreeSurfer on the real data from ADNI and ICBM.

C. Quantitative comparisons on ADNI data

In the third experiment, we applied both our method and FreeSurfer to the 100 MR images from ADNI. To compare the performance of our method and FreeSurfer in detecting group differences, we parcellated cortical surfaces into gyral regions and use the mean thickness of each gyrus as the variable for statistical tests. As described in the previous experiment, we used the gyral labels on GM surfaces generated by FreeSurfer to parcellate our surfaces via nearest neighbor projection.

For surfaces generated by both methods, the thickness of each vertex on the GM surface was computed as its distance to the corresponding WM surface. For each gyrus, we computed the average gray matter thickness and tested group differences between NC and AD using two tailed t-tests. As an illustration of the thickness differences across groups, we plotted in Fig. 17 the mean thickness of gyral regions from the NC and AD group. By comparing the thickness of corresponding gyrus from the two groups, we can clearly see that the AD group have thinner cortex throughout the brain. For the results produced by our method and FreeSurfer, the p-values from t-tests were mapped onto the corresponding gyrus of the left hemisphere (LH) and right hemisphere (RH), and plotted in Fig. 18. Note that p-values larger than 0.01 were considered insignificant and plotted in light gray. We can see that both methods are very effective in detecting differences between the NC and AD groups on both hemispheres, whereas the degrees of difference vary on different gyri. For the left hemisphere, results from our method produces more significant p-values on 33 of the 34 regions. For the right hemisphere, our results generate more significant p-values on 30 of the 34 regions.

For the two-tailed t-test on the mean thickness of each gyrus, we also performed power analysis [69] to compare the power of thickness features generated by our method and FreeSurfer in detecting group differences. The power of the t-test is the probability that the null hypothesis will be rejected if it is false. The type I error, i.e., the p-value, was set to 0.01. For each method, the effective size at a gyrus was computed as $\frac{MT_{NC} - MT_{AD}}{\sigma}$ [69], where MT_{NC} and MT_{AD} were the mean thickness of the NC and AD group, respectively, and σ was the pooled standard deviation. The power was computed with the software tool *PS* [70]. For both our method and FreeSurfer, we counted the number of gyri in each hemisphere with power greater than four thresholds 0.99, 0.95, 0.90, 0.85. The results are listed in Table III. We can see from the power analysis that our method is able to generate more sensitive thickness features on both hemispheres for all thresholds. From a study design point of view, this shows the thickness features derived from our method has the potential of reducing sample sizes in experiments for AD research.

D. Quantitative comparisons on ICBM data

In the fourth experiment, we tested on MR images of 100 subjects from the ICBM database. We split the 100 subjects into young and elderly groups using an age threshold of 48 years such that each group had 50 subjects. For each subject, we downloaded two scans for our experiments. For the first scan of each subject, we applied our method and FreeSurfer for comparison. The repeat scan of each subject will be used to demonstrate the reproducibility of our method.

Using surfaces generated by both methods on the first scan of each subject, we tested for group differences between the young and elderly groups and compared the statistical power of our method and FreeSurfer. Using the same method described in VI-B, we projected FreeSurfer labels onto the surfaces generated by our method and applied two-tailed t-tests to the mean thickness of each gyrus from the young and elderly groups. The mean thickness of gyral regions obtained from our method were plotted in Fig. 19, where we can see that the elderly group have lower thickness throughout the cortex. The p-value maps generated by our method and FreeSurfer were plotted in Fig. 20, where the same colormap as in Fig. 18 was used and p-values over 0.01 were plotted in light gray. For each hemisphere, t-tests with thickness measures generated by our method have more significant p-values than FreeSurfer on 33 of the 34 gyri.

Similar to the experiment on ADNI data, we also performed power analysis for the mean thickness measure generated by our method and FreeSurfer at each gyrus. For both methods, the p-value was set to 0.01, and the effective size was computed as $\frac{MT_Y - MT_O}{\sigma}$, where MT_Y and MT_O were the mean thickness of the young and elderly groups, and σ was the pooled standard deviation. For each method, we counted the number of gyri on each hemisphere with power exceeding the thresholds. The results were listed in Table IV. On this data set, we can see our method is able to generate a much larger number of sensitive thickness features than FreeSurfer. This demonstrates that our method achieves better performance on both the ADNI and ICBM datasets with very different demographic distributions.

To test the reproducibility of our method, we also applied our method to the repeat scan from each of the 100 subjects in the ICBM dataset. To parcellate each GM surface reconstructed from a repeat scan into gyral regions, we rigidly aligned it with the corresponding GM surface from the first scan and used the nearest neighbor projection as described in VI-B to generate the gyral labels. For the results obtained from the repeat scans, the same power analysis was applied to the mean thickness of each gyrus from the young and elderly group. The results were listed in Table V. Compared with the results from the first scans, we can see that our method is able to successfully generate very similar and large numbers of sensitive thickness features in the repeat scans, which demonstrates the reproducibility of our method.

VII. Conclusions

In this paper we have developed a novel system for automated cortical reconstruction from T1-weighted MR images. We have demonstrated that our system is able to generate high quality surfaces with much reduced computational cost when compared with FreeSurfer. The unified approach presented in this work for the analysis of geometric and topological structures, while developed in the context of cortical reconstruction, is general and applicable for other medical image analysis problems.

In this work, we focused on the reconstruction of the WM and GM surfaces. There are also considerable interests in obtaining a central cortical surface from MR images as a balanced representation of the WM and GM surfaces [7]. There are two different approaches we can follow to achieve this goal. Using the reconstructed WM and GM surfaces, a surface evolution algorithm can be developed as in [7] to evolve the WM surface to the middle of the GM tissue. We can also take a shape analysis approach to find the central cortical surface. In our recent work, we developed an intrinsic surface mapping method that can evolve the WM and GM surface toward each other in a high dimensional embedding space by optimizing their conformal metrics [43]. By establishing one-to-one correspondences between the WM and GM surface with this intrinsic mapping algorithm, we can compute the central cortical surface as the mean shape of the WM and GM surface. Using the one-to-one

map between the WM and GM surfaces, we can also develop a more robust method to compute cortical thickness than using the distance transform of the WM surface, which can result in possible underestimation of thickness in the extreme case of neighboring sulcal banks with highly uneven thickness.

In future work, we will also perform more extensive validations on data from a wide range of neuroimaging studies, and extend the system to the consistent reconstruction of surfaces on data from longitudinal studies [11], [71].

Acknowledgments

Data collection and sharing from ADNI was funded by National Institutes of Health Grant U01 AG024904. ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Abbott; Alzheimers Association; Alzheimers Drug Discovery Foundation; Amorfix Life Sciences Ltd.; AstraZeneca; Bayer Health-Care; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals Inc.; Eli Lilly and Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of California, Los Angeles. Data collection and sharing from ADNI was also supported by NIH grants P30 AG010129 and K01 AG030514.

This work was in part supported by the National Institute of Health (NIH) under Grant K01EB013633 and Grant 5P41RR013642.

References

1. Mangin JF, Frouin V, Bloch I, et al. From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. *J MATH IMAGING VIS.* 1995; 5(4):297–318.
2. Carman GJ, Drury HA, Van Essen DC. Computational methods for reconstructing and unfolding the cerebral cortex. *Cerebral cortex.* 1995; 5:506–517.
3. Davatzikos C, Bryan RN. Using a deformable surface model to obtain a shape representation of the cortex. *IEEE Trans Med Imag.* 1996; 15:785–795.
4. MacDonald, D. PhD dissertation. McGill Univ; Canada: 1998. A method for identifying geometrically simple surfaces from three dimensional images.
5. Dale AM, Fischl B, Sereno MI. Cortical surface-based analysis i: segmentation and surface reconstruction. *NeuroImage.* 1999; 9:179–194. [PubMed: 9931268]
6. Shattuck D, Leahy R. BrainSuite: An automated cortical surface identification tool. *Med Image Anal.* 2002; 8(2):129–142. [PubMed: 12045000]
7. Han X, Pham DL, Tosun D, et al. CRUISE: Cortical reconstruction using implicit surface evolution. *NeuroImage.* 2004; 23:997–1012. [PubMed: 15528100]
8. Kim JS, Singh V, Lee JK, Lerch J, Ad-Dab'bagh Y, MacDonald D, Lee JM, Kim SI, Evans AC. Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a laplacian map and partial volume effect classification. *NeuroImage.* 2005; 27:210–221. [PubMed: 15896981]
9. Xue H, Srinivasan L, Jiang S, Rutherford M, Edwards AD, Rueckert D, Hajnal JV. Automatic segmentation and reconstruction of the cortex from neonatal MRI. *NeuroImage.* 2007; 38(3):461–477. [PubMed: 17888685]
10. Liu T, Nie J, Tarokh A, Guo L, STW. Reconstruction of central cortical surface from brain MRI images: method and application. *NeuroImage.* 2008; 40(3):991–1002. [PubMed: 18289879]
11. Li G, Nie J, Wu G, Wang Y, Shen D. Consistent reconstruction of cortical surfaces from longitudinal brain mr images. *NeuroImage.* 2012; 59(4):3805– 3820. [PubMed: 22119005]

12. Sled J, Zijdenbos A, Evans A. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE Trans Med Imag.* 1998; 17(1):87–97.
13. Christensen GE, Rabbitt RD, Miller MI. Deformable templates using large deformation kinematics. *IEEE Trans Imag Process.* 1996; 5(10):1435–1447.
14. Joshi S, Miller MI. Landmark matching via large deformation diffeomorphisms. *IEEE Trans Imag Process.* 2000; 9(8):1357–1370.
15. Shen D, Davatzikos C. HAMMER: hierarchical attribute matching mechanism for elastic registration. *IEEE Trans Med Imag.* 2002; 21(11):1421–39.
16. Avants BB, Epstein CL, Grossman M, Gee JC. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Med Image Anal.* 2008; 12(1):26–42. [PubMed: 17659998]
17. Smith S. Fast robust automated brain extraction. *Hum Brain Mapp.* 2002; 17(3):143–55. [PubMed: 12391568]
18. Ségonne F, Dale A, Busa E, Glessner M, Salat D, Hahn H, Fischl B. Robust brain extraction across datasets and comparison with publicly available methods. *NeuroImage.* 2004; 22(3):1660–75.
19. Iglesias J, Liu CY, Thompson P, Tu Z. Robust brain extraction across datasets and comparison with publicly available methods. *IEEE Trans Med Imag.* 2011; 30(9):1617–34.
20. Leemput KV, Maes F, Vandermeulen D, Suetens P. Automated model-based tissue classification of MR images of the brain. *IEEE Trans Med Imag.* 1999; 18(10):897–908.
21. Zhang Y, Brady M, Smith S. Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. *IEEE Trans Med Imag.* 2001; 20(1):45–57.
22. Shattuck D, Leahy R. Automated graph-based analysis and correction of cortical volume topology. *IEEE Trans Med Imag.* 2001; 20(11):1167–1177.
23. Bazin P, Pham D. Topology-preserving tissue classification of magnetic resonance brain images. *IEEE Trans Med Imag.* 2007; 26(4):487–96.
24. Han X, Xu C, Prince J. A topology preserving level set method for geometric deformable models. *IEEE Trans Pattern Anal Machine Intell.* Jun; 2003 25(6):755–768.
25. Bazin P, Pham D. Topology correction using fast marching methods and its application to brain segmentation. *Proc MICCAI.* 2:484–91.
26. Han X, Xu C, Braga-Neto U, Prince JL. Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Trans Med Imag.* 2002; 21(2):109–121.
27. Chen L, Wagenknecht G. Topological correction of volumetric binary brain segmentation using a multiscale algorithm. *Proc ISBI.* 2007:1308–11.
28. Ségonne F, Grimson E, Fischl B. A genetic algorithm for the topology correction of cortical surfaces. *Proc IPMI.* 2005; 19:393–405.
29. Yotter R, Dahnke R, Thompson P, Gaser C. Topological correction of brain surface meshes using spherical harmonics. *Hum Brain Mapp.* 2011; 32(7):1109–24. [PubMed: 20665722]
30. Latecki LJ. 3D well-composed pictures. *Graph Models Image Process.* 1997; 59(3):164–172.
31. Shi Y, Lai R, Morra J, Dinov I, Thompson P, Toga A. Robust surface reconstruction via Laplace-Beltrami eigen-projection and boundary deformation. *IEEE Trans Med Imag.* 2010; 29(12):2009–2022.
32. Reuter M, Wolter F, Peinecke N. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design.* 2006; 38:342–366.
33. Qiu A, Bitouk D, Miller MI. Smooth functional and structural maps on the neocortex via orthonormal bases of the Laplace-Beltrami operator. *IEEE Trans Med Imag.* 2006; 25(10):1296–1306.
34. Rostamov RM. Laplace-beltrami eigenfunctions for deformation invariant shape representation. *Eurographics Symposium on Geometry Processing (SGP).* 2007:225–233.
35. Niethammer M, Reuter M, Wolter F-E, et al. Global medical shape analysis using the Laplace-Beltrami spectrum. *Proc MICCAI.* 2007:850–857.
36. Vallet B, Lvy B. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum.* 2008; 27(2):251–260.

37. Shi Y, Lai R, Krishna S, et al. Anisotropic Laplace-Beltrami eigenmaps: Bridging Reeb graphs and skeletons. *Proc MMBIA*. 2008:1–7.
38. Shi Y, Lai R, Kern K, Sicotte N, Dinov I, Toga AW. Harmonic surface mapping with Laplace-Beltrami eigenmaps. *Proc MICCAI*. 2008; 2:147–154.
39. Shi Y, Morra J, Thompson PM, Toga AW. Inverse-consistent surface mapping with Laplace-Beltrami eigen-features. *Proc IPMI*. 2009:467–478.
40. Lai R, Shi Y, Dinov I, Chan TF, Toga AW. Laplace-beltrami nodal counts: a new signature for 3d shape analysis. *Proc ISBI*. 2009:694–697.
41. Lai R, Shi Y, Scheibel K, Fears S, Woods R, Toga A, Chan T. Metric-induced optimal embedding for intrinsic 3D shape analysis. *Proc CVPR*. 2010:2871–2878.
42. Shi Y, Lai R, Toga A. CoRPORATE: Cortical reconstruction by pruning outliers with Reeb analysis and topology-preserving evolution. *Proc IPMI*. 2011:233–244.
43. Shi Y, Lai R, Gill R, Pelletier D, Mohr D, Sicotte N, Toga A. Conformal metric optimization on surface (CMOS) for deformation and mapping in Laplace-Beltrami embedding space. *Proc MICCAI*. 2011; 2:327–334.
44. Reeb G. Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction nemérique. *Comptes Rendus Acad Sciences*. 1946; 222:847–849.
45. Shinagawa Y, Kunii TL. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics & Applications*. 1991; 11(6):44–51.
46. Cole-McLaughlin K, Edelsbrunner H, Harer J, Natarajan V, Pascucci V. Loops in Reeb graphs of 2-manifolds. *Discrete Comput Geom*. 2004; 32(2):231–244.
47. Pascucci V, Scorzelli G, Bremer PT, Mascarenhas A. Robust on-line computation of Reeb graphs: simplicity and speed. *ACM Trans Graph*. 2007; 26(3):58, 1–9.
48. Wood Z, Hoppe H, Desbrun M, Schroder P. Removing excess topology from isosurfaces. *ACM Trans on Graph*. 2004; 23(2):190–208.
49. Patanè G, Spagnuolo M, Falcidieno B. A minimal contouring approach to the computation of the reeb graph. *IEEE Trans Vis and Comp Graph*. 2009; 15(4):583–595.
50. Frangi A, Niessen W, Hoogeveen R, van Walsum T, Viergever M. Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Trans Med Imag*. 1999; 18(10):946–956.
51. Sato Y, Westin CF, Bhalerao A, Nakajima S, Shiraga N, Tamura S, Kikinis R. Tissue classification based on 3D local intensity structure for volume rendering. *IEEE Trans Vis and Comp Graph*. 2000; 6(2):160–180.
52. Fischl B, Sereno MI, Dale AM. Cortical surface-based analysis ii: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*. 1999; 9(2):195–207. [PubMed: 9931269]
53. Shattuck D, Mirza M, Adisetiyo V, et al. Construction of a 3D probabilistic atlas of human brain structures. *NeuroImage*. 2008; 39(3):1064–1080. [PubMed: 18037310]
54. Klein A, Andersson J, Ardekani BA, Ashburner J, Avants B, Chiang MC, Christensen GE, Collins DL, Gee J, Hellier P, Song JH, Jenkinson M, Lepage C, Rueckert D, Thompson P, Vercauteren T, Woods RP, Mann JJ, Parsey RV. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *NeuroImage*. 2009; 46(3):786–802. [PubMed: 19195496]
55. Siddiqi K, Bouix S, Tannebaum A, Zuker S. Hamilton-Jacobi skeletons. *Int'l Journal of Computer Vision*. 2002; 48(3):215–231.
56. Shi Y, Karl WC. Real-time tracking using level sets. *Proc CVPR*. 2005; 2:34–41.
57. Shi Y, Karl WC. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Trans Image Processing*. 2008; 17(5):645–657.
58. Silvester, PP.; Ferrari, RL. *Finite Elements for Electrical Engineers*. Cambridge University Press; 1996.
59. Uhlenbeck K. Generic properties of eigenfunctions. *Amer J of Math*. 1976; 98(4):1059–1078.
60. Kimmel R, Sethian JA. Computing geodesic paths on manifolds. *Proc Natl Acad Sci USA*. 1998; 95(15):8431–8435. [PubMed: 9671694]
61. Aubert-Broche B, Griffin M, Pike G, Evans A, Collins D. Twenty new digital brain phantoms for creation of validation image data bases. *IEEE Trans Med Imag*. 2006; 25(11):1410–6.

62. Mueller S, Weiner M, Thal L, et al. The Alzheimer's disease neuroimaging initiative. *Clin North Am.* 2005; 15:869–877. xi–xii.
63. Mazziotta JC, Toga AW, Evans AC, et al. A probabilistic atlas and reference system for the human brain: international consortium for brain mapping. *Philos Trans R Soc Lond B Biol Sci.* 2001; 356:1293–1322. [PubMed: 11545704]
64. Leung K, Parker D, Cunha A, et al. IRMA: An image registration meta-algorithm evaluating alternative algorithms with multiple metrics. *Proc Int Conf on Scientific and Statistical Data Base Management.* 2008:612–617.
65. Dinov I, Van Horn J, Lozev K, et al. Efficient, distributed and interactive neuroimaging data analysis using the LONI Pipeline. *Front Neuroinform.* 2009; 3
66. Brainweb: Simulated Brain Database. [online] Available: <http://www.bic.mni.mcgill.ca/brainweb>
67. Miller MI, Massie AB, Ratnanather J, Botteron KN, Csernansky JG. Bayesian construction of geometrically based cortical thickness metrics. *NeuroImage.* 2000; 12(6):676– 687. [PubMed: 11112399]
68. Sethian J. A fast marching level set method for monotonically advancing fronts. *Proc Nat Acad Sci.* 1996; 93(4):1591–1595. [PubMed: 11607632]
69. Cohen J. Quantitative methods in psychology: a power primer. *Psychological Bulletin.* 1992; 112(1):155–59. [PubMed: 19565683]
70. Dupont W, Plummer W. Power and sample size calculations: A review and computer program. *Controlled Clinical Trials.* 1990; 11:116–28. [PubMed: 2161310]
71. Reuter M, Schmansky NJ, Rosas HD, Fischl B. Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage.* 2012 in print. 10.1016/j.neuroimage.2012.02.084

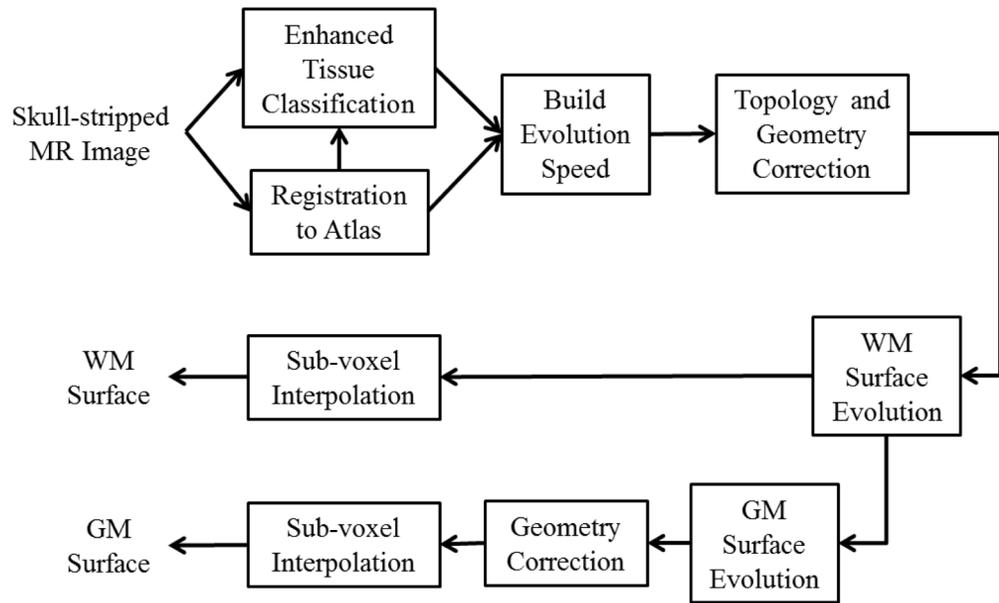


Fig. 1.
An overview of our cortical reconstruction system.

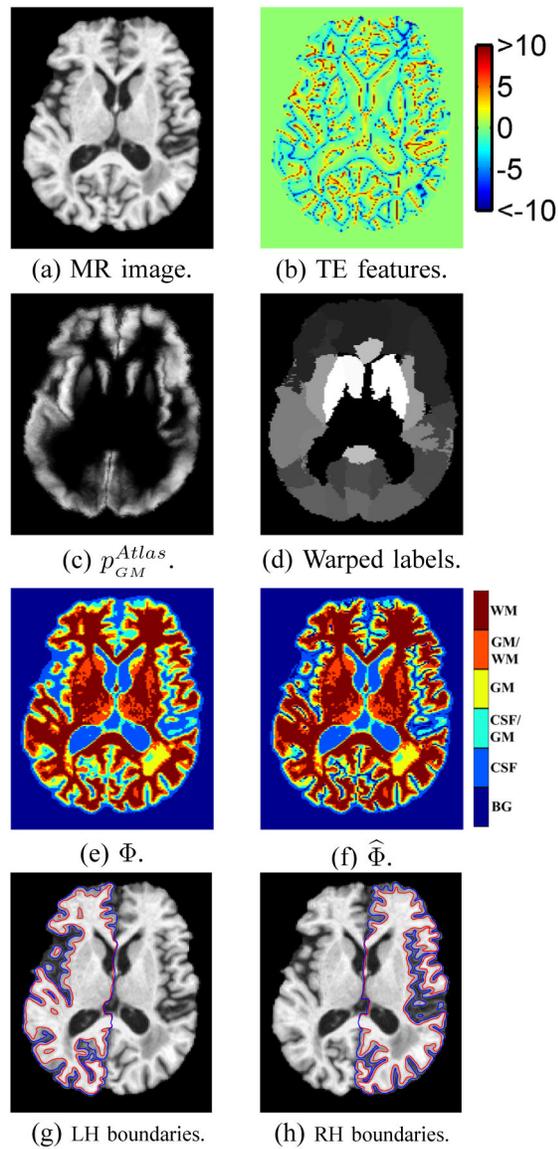


Fig. 2. Tissue enhancement with Hessian features and atlas labels. Using enhanced tissue maps and atlas labels, evolution speeds are designed to reconstruct the WM (red) and GM (blue) boundaries of the left hemisphere (LH) and right hemisphere as plotted in (g) and (h).

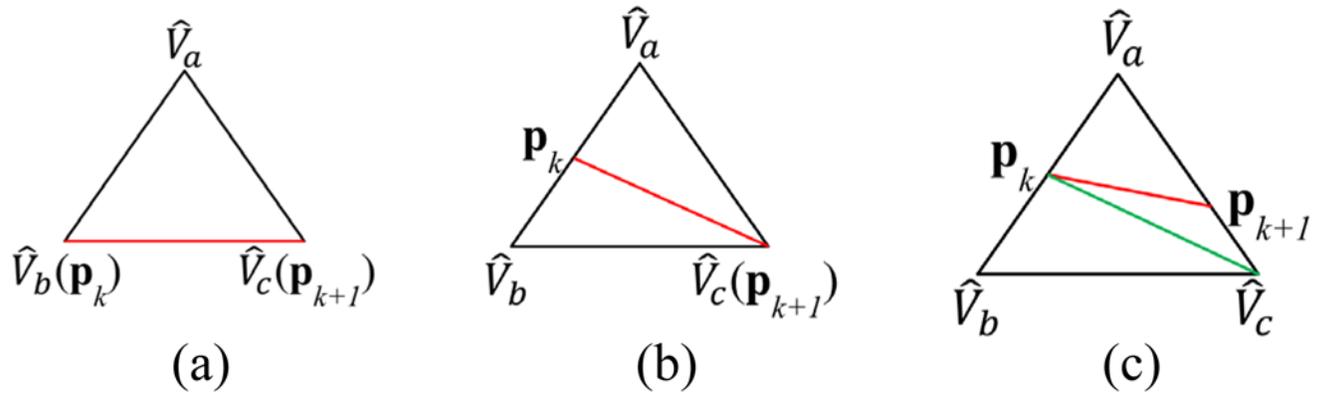


Fig. 3. Mesh augmentation by splitting triangles. Original edge: black. Level contour: red. Inserted edge: green. (a) Keep the triangle. (b) Split into two triangles. (c) Split into three triangles.

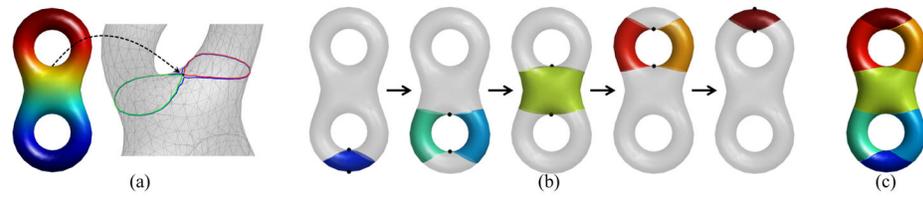


Fig. 4. Reeb graph construction of a double torus using a LB eigenfunction. (a) An illustration of the level contours in the neighborhood of a saddle point. (b) An illustration of the arc construction process in building the Reeb graph. (c) The final partition of the surface with the Reeb graph.

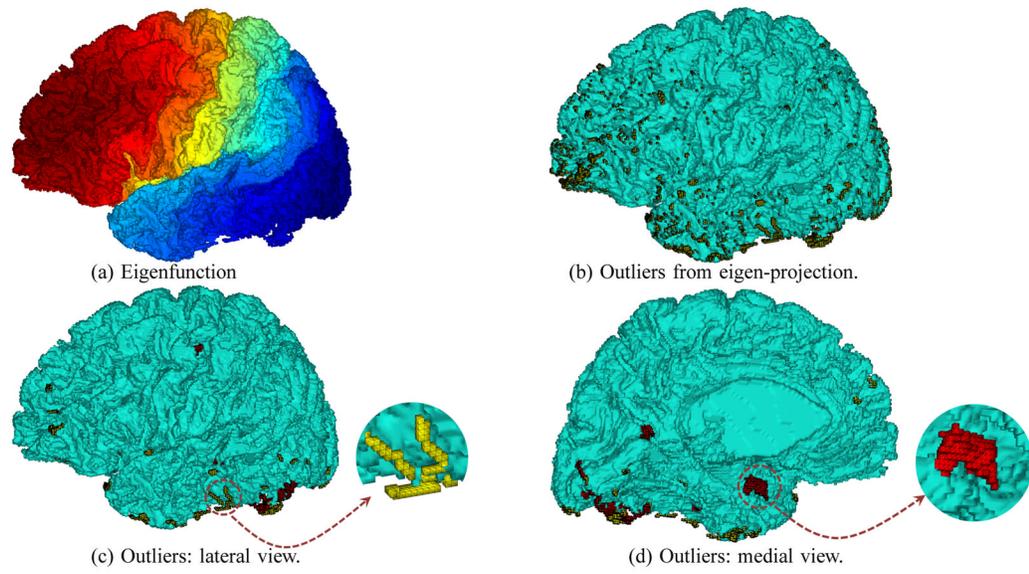


Fig. 5. Geometric and topological outlier detection with the Reeb graph of the LB eigenfunction. Zoomed views of geometric and topological outliers are plotted in (c) and (d), respectively. Yellow: Geometric outliers. Red: Topological outliers.

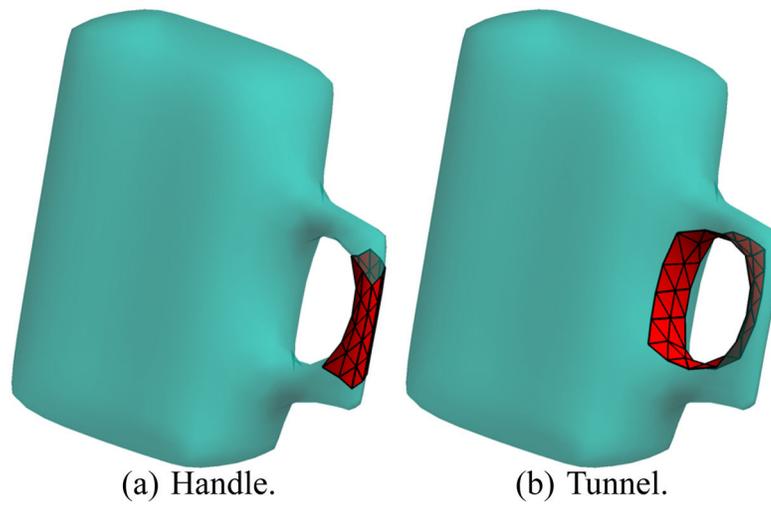


Fig. 6. A Reeb graph can capture a topological outlier with an arc in the form of a handle (a) or tunnel (b).

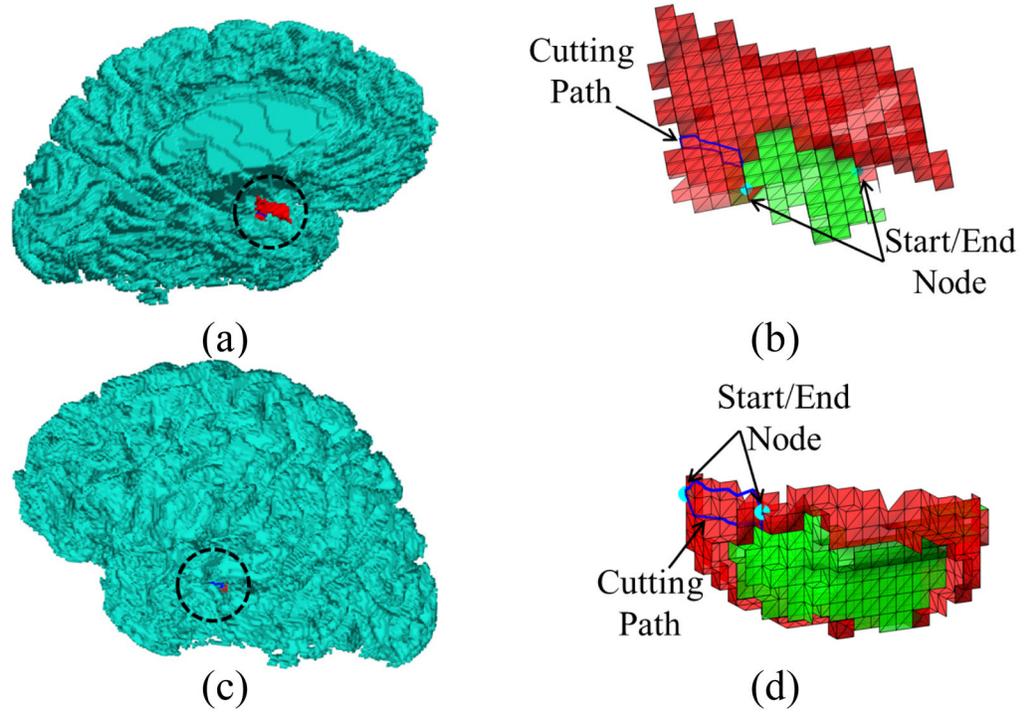


Fig. 7. Topological analysis of handles and tunnels detected with Reeb analysis. (a) A handle detected with Reeb analysis. (b) Red: arc of the Reeb graph. Green: filling voxels. (c) A tunnel detected by Reeb analysis. (d) Red: arc of the Reeb graph. Green: filling voxels.

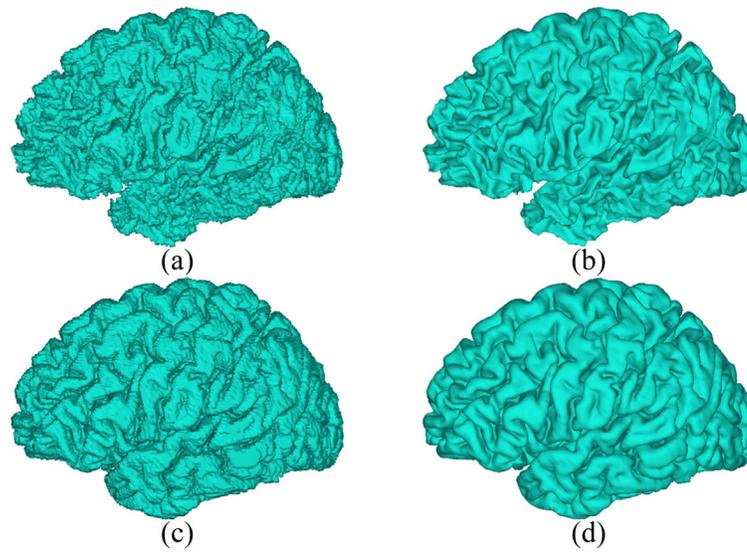


Fig. 8. The WM and GM surface before and after sub-voxel calculations. (a) WM boundary after removal of outliers. (b) WM surface with sub-voxel accuracy. (c) GM boundary after removal of outliers. (d) GM surface with sub-voxel accuracy.

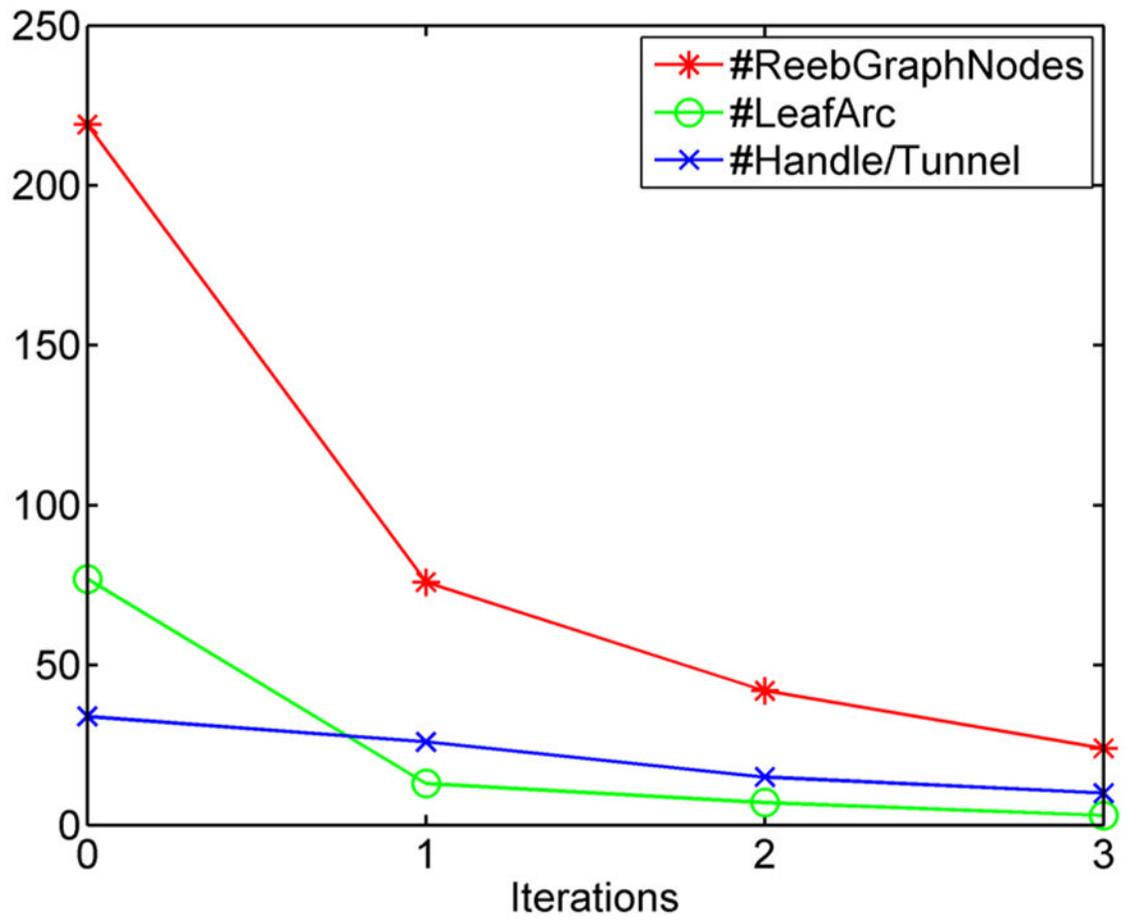


Fig. 9.
The iterative process of outlier removal with Reeb analysis.

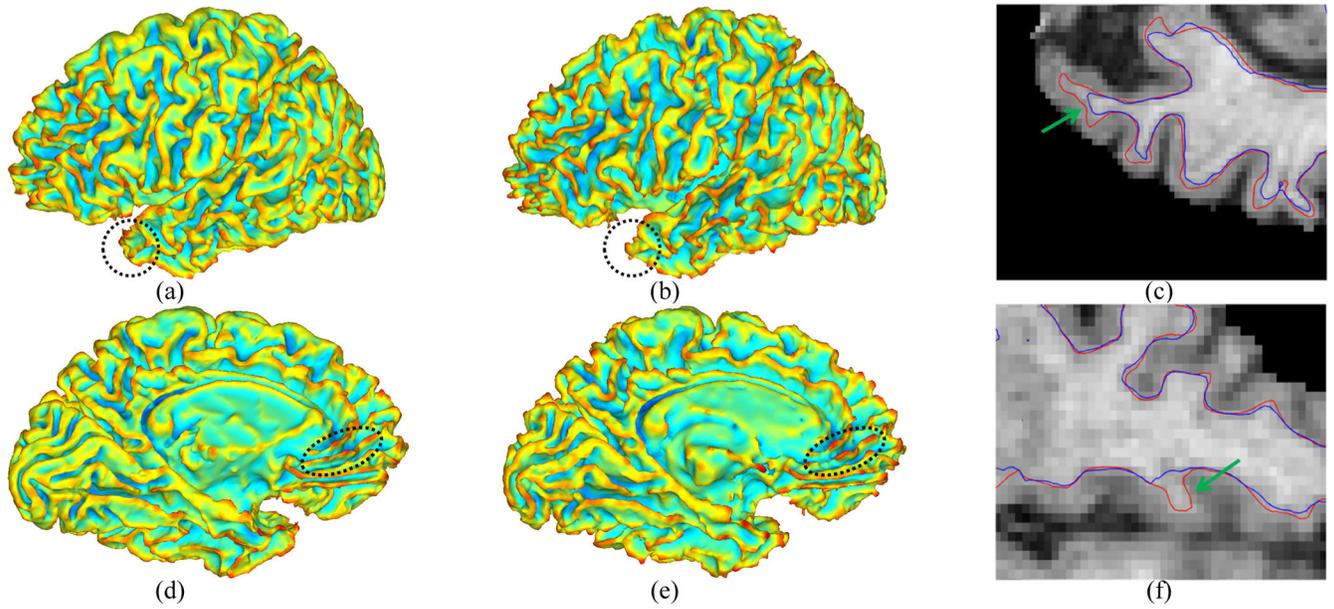


Fig. 10. Comparison of WM surfaces reconstructed by our method and FreeSurfer. Surfaces are colored by their mean curvature and plotted in both lateral and medial views. (a)(d) Our result. (b)(e) FreeSurfer result. (c)(f) Intersection of an image slice with circled regions in (a)(b) and (d)(e), respectively. Red contour: our result. Blue contour: FreeSurfer result. Green arrows in (c) and (f) highlight the differences.

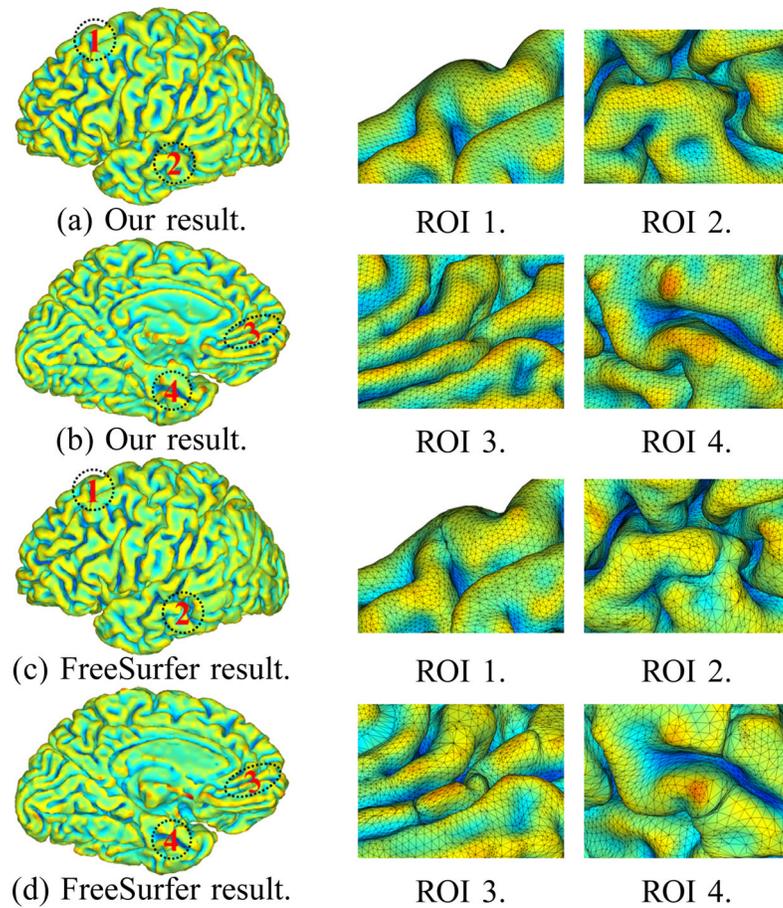


Fig. 11. Comparison of the GM surface reconstructed by our method and FreeSurfer. (a)(b) Lateral and medial views of our result. (c)(d) Lateral and medial views of FreeSurfer result. The mesh structures of four highlighted ROIs are plotted to demonstrate differences between these two results.

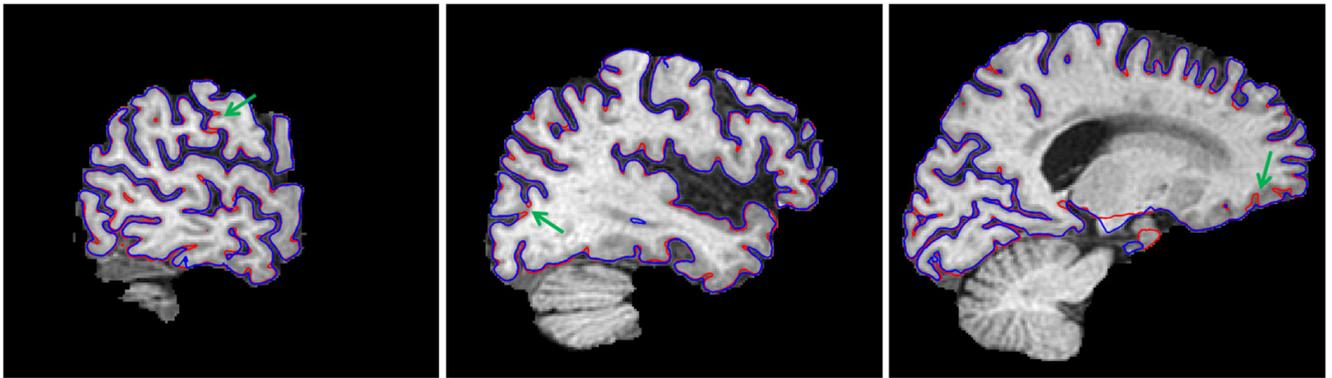


Fig. 12. Intersection of image slices with GM surfaces reconstructed with our method (red contours) and FreeSurfer (blue contours). Green arrows point to locations where our surface can better capture deep sulcal regions.

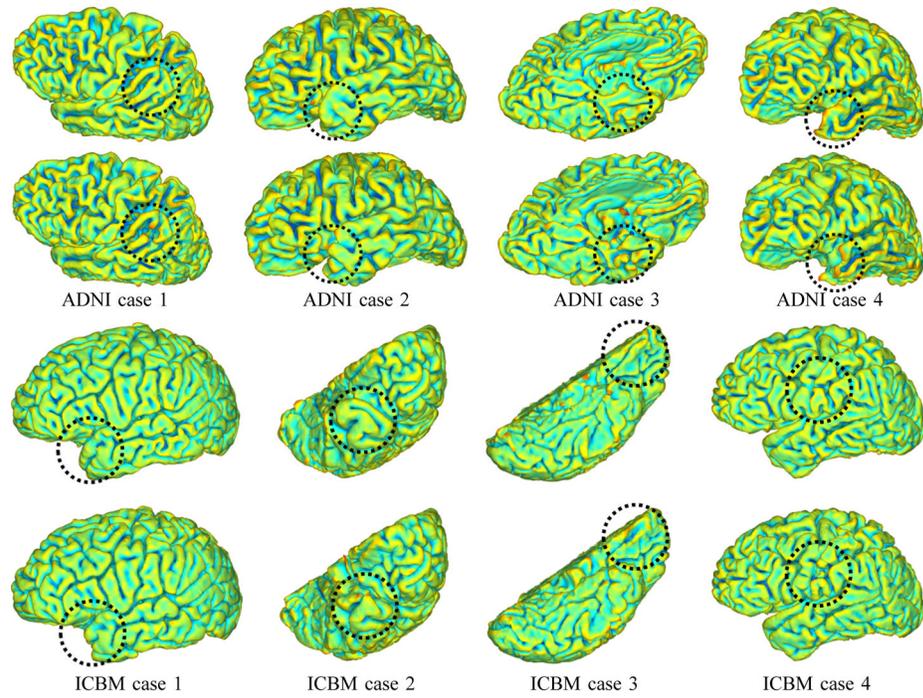


Fig. 13. Comparison of cortical reconstruction results on four ADNI and four ICBM cases. For each case, the result from our method is plotted on the top, and the result from FreeSurfer is plotted on the bottom. Different views are selected for different subjects to better illustrate differences in regions highlighted with dashed circles.

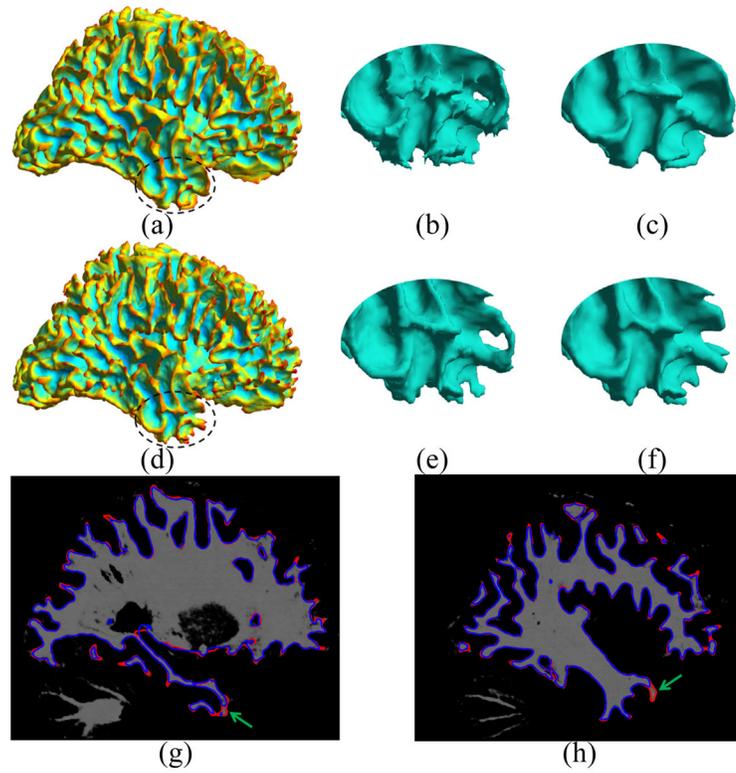


Fig. 14.

Comparison of our method and FreeSurfer on a simulated image with known WM tissue map. (a) The WM surface reconstructed by our method. (b)(c) A zoomed view of the circled region in (a) before topology correction and after final reconstruction. (d) The WM surface reconstructed by FreeSurfer. (e)(f) A zoomed view of the circled region in (d) before topology correction and after final reconstruction. (g)(h) Intersection of reconstructed surfaces with the true WM tissue map. Red contour: our method; Blue contour: FreeSurfer.

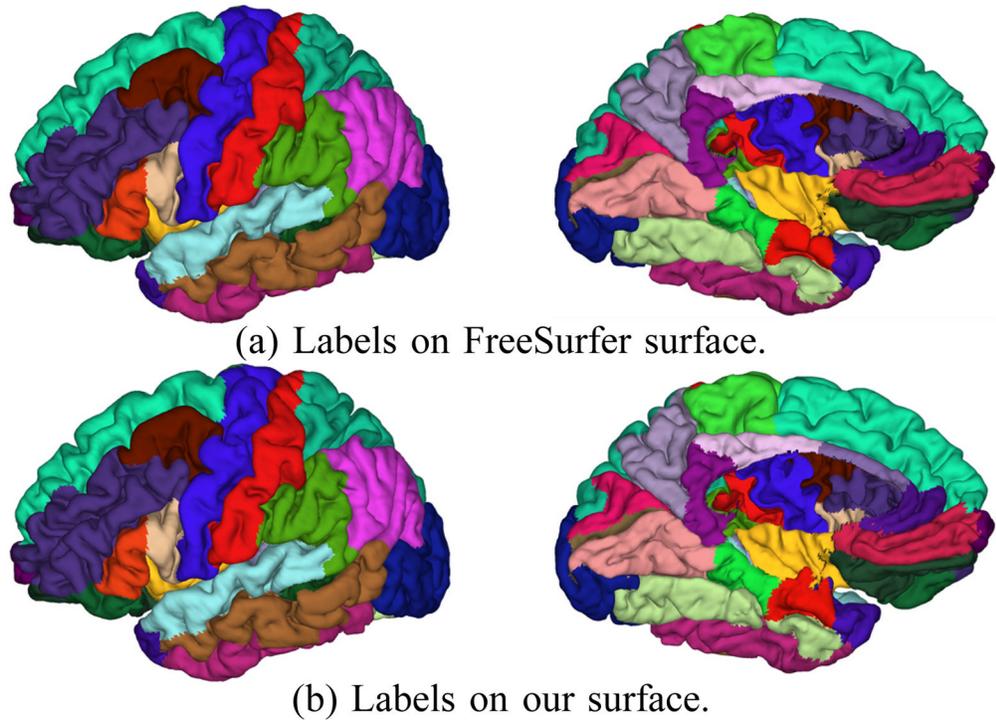


Fig. 15. Projection of FreeSurfer gyral labels onto the GM surface reconstructed by our method.

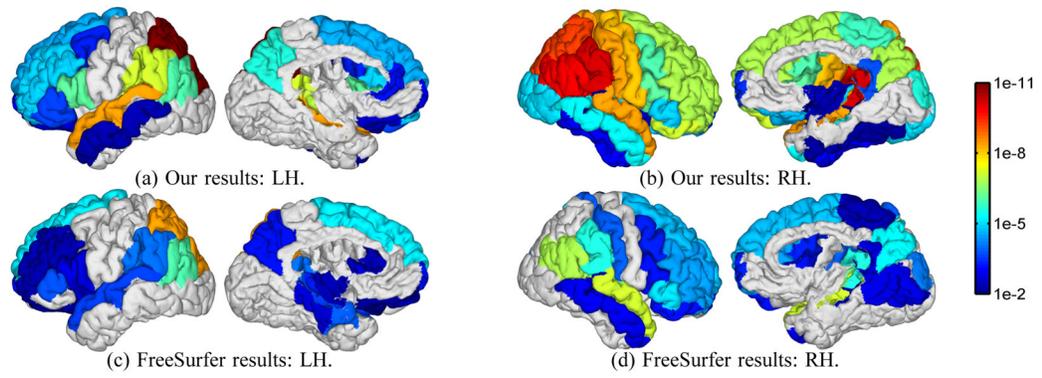


Fig. 16. Gyrus-based map of p-values from the testing of longitudinal atrophy on LH and RH surfaces from our results (a, b) and FreeSurfer results (c, d).

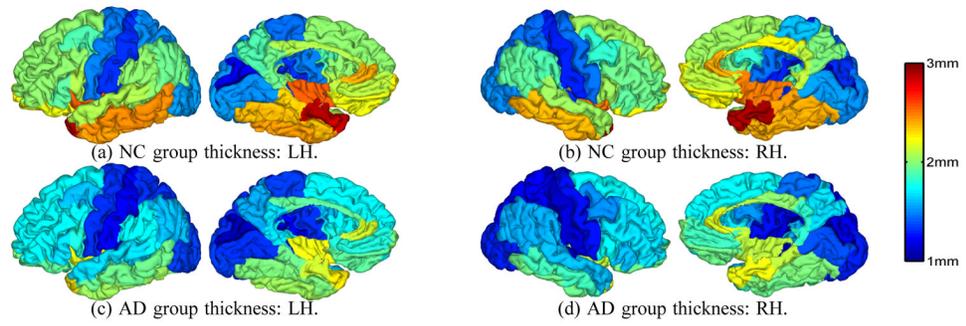


Fig. 17. Mean thickness of gyral regions on the left and right hemisphere of the NC and AD group.

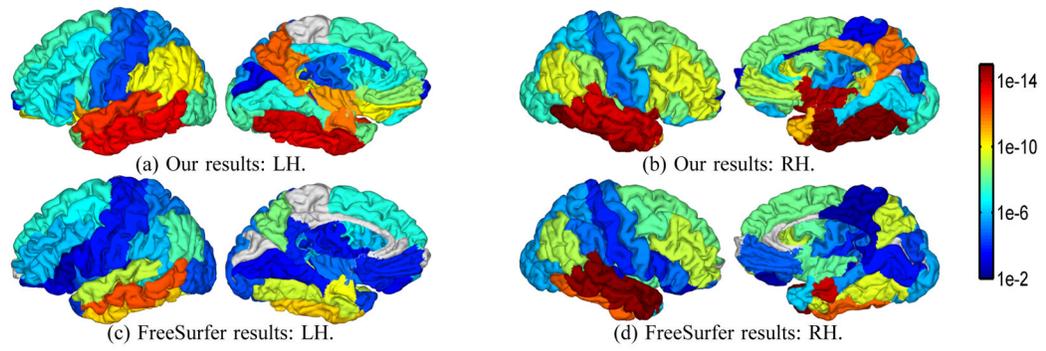


Fig. 18. Gyrus-based map of p-values from the testing of NC versus AD group differences on LH and RH surfaces from our results (a,b), and FreeSurfer results (c, d).

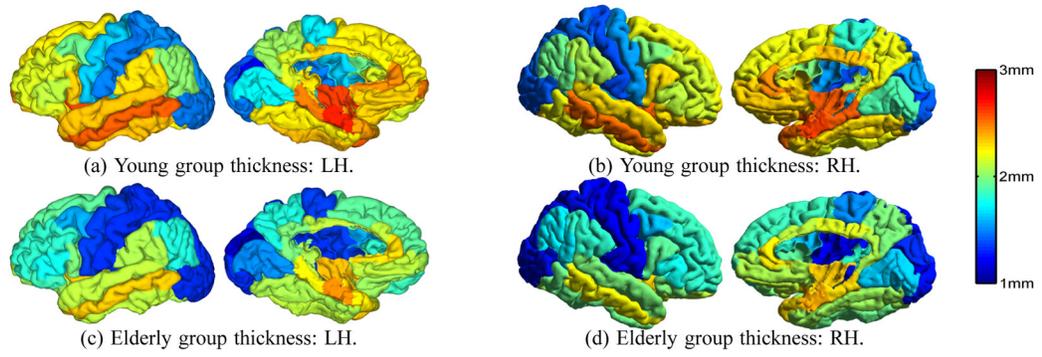


Fig. 19.
Gyrus-based thickness map of the young and elderly group.

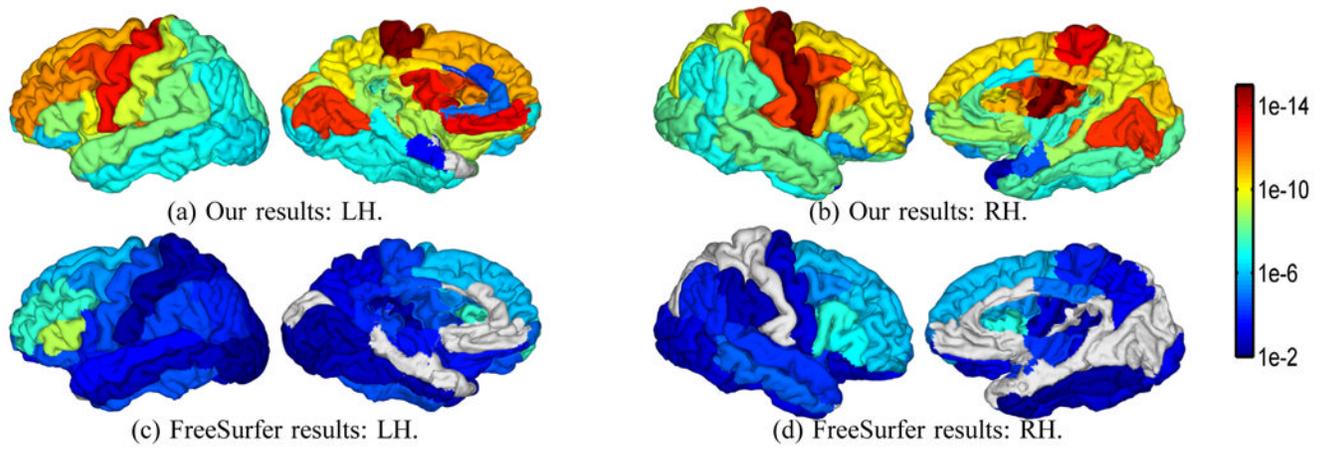


Fig. 20. Gyrus-based map of p-values from the testing of young and elderly group differences on LH and RH surfaces from our results (a, b) and FreeSurfer results (c, d).

TABLE I

Ridge/Valley Detection Algorithm

R : Region to be thinned.
 R_F : ridge/valley detection feature.

Initialize a heap \mathcal{H} with voxels on the boundary of R using R_F as the sorting key.

do

$P = \text{PopHeap}(\mathcal{H})$.

 If P is a simple point but not an end point [55]

 Remove P from R .

 Insert neighbors of P in R to \mathcal{H} if they are simple points.

 End

while \mathcal{H} is not empty.

The voxels left in R is the ridge/valley set.

TABLE II

Reeb Graph Arc Growing

R_f : Initial set of vertices in the region.

L_V : Label vector initialized to be 1 for vertices in S and zero for other vertices in V .

L_T : Label vector initialized to be zero for all triangles in T .

\mathcal{T} : Triangle set initialized to be empty.

do

 Remove the first vertex \mathbf{x} from R_f .

 For each triangle $T_i \in \mathcal{N}(\mathbf{x})$ with label $L_T = 0$

 If all vertices of T_i have label $L_V = 1$

 Add T_i to \mathcal{T}

 Set the label $L_T = 1$ at T_i .

 End

 End

 If $ArcLabel(\mathbf{x}) = -1$

 For each vertex $\mathbf{y} \in \mathcal{N}(\mathbf{x})$

 If $f(\mathbf{y}) < f(\mathbf{x})$ and $L_V = 0$

 Add \mathbf{y} to R_f .

 Set the label $L_V = 1$ at \mathbf{y} .

 End

 End

 Else

 Record current arc label as $ArcLabel(\mathbf{x})$.

 End

while R_f is not empty.

For vertices in the triangles of \mathcal{T} , reset the ArcLabel to -1 .

TABLE III

Power analysis of the ADNI experiment: number of thickness features from our method and FreeSurfer with power above thresholds.

Power Threshold	Our Method LH	FreeSurfer LH	Our Method RH	FreeSurfer RH
0.99	25	12	28	14
0.95	27	15	29	21
0.90	30	21	30	25
0.85	32	22	31	25

TABLE IV

Power analysis on first scans of ICBM dataset: number of thickness features from our method and FreeSurfer with power above thresholds.

Power Threshold	Our Method LH	FreeSurfer LH	Our Method RH	FreeSurfer RH
0.99	30	3	30	4
0.95	32	9	33	8
0.90	32	14	33	12
0.85	33	18	33	12

TABLE V

Power analysis on repeat scans of ICBM dataset: number of thickness features from our method with power above thresholds.

Power Threshold	Our Method LH	Our Method RH
0.99	29	27
0.95	31	32
0.90	32	32
0.85	32	32