

Automated Graph-Based Analysis and Correction of Cortical Volume Topology

David W. Shattuck* and Richard M. Leahy

Abstract—The human cerebral cortex is topologically equivalent to a sheet and can be considered topologically spherical if it is closed at the brain stem. Low-level segmentation of magnetic resonance (MR) imagery typically produces cerebral volumes whose tessellations are not topologically spherical. We present a novel algorithm that analyzes and constrains the topology of a volumetric object. Graphs are formed that represent the connectivity of voxel segments in the foreground and background of the image. These graphs are analyzed and minimal corrections to the volume are made prior to tessellation. We apply the algorithm to a simple test object and to cerebral white matter masks generated by a low-level tissue identification sequence. We tessellate the resulting objects using the marching cubes algorithm and verify their topology by computing their Euler characteristics. A key benefit of the algorithm is that it localizes the change to a volume to the specific areas of its topological defects.

Index Terms—Magnetic resonance imaging, topological correction, topology, segmentation.

I. INTRODUCTION

GOMETRIC information about the cerebral cortex is important in various neuroimaging applications including intersubject registration and comparison, image reconstruction, and visualization techniques. The field of brain mapping has focused much attention on producing flat and other two-dimensional (2-D) representations of the cerebral cortical surface in order to produce coordinate systems in which brain geometries and other properties may be compared [1]–[9]. These methods inherently rely on an important topological property of the cerebral cortex—that it is a single sheet of connected tissue enclosing the telencephalon. By closing the cortical surface at the brainstem we can describe the cerebral cortex as having the topology of a sphere. This topology provides us with an important ability—it guarantees the existence of invertible one-to-one maps between cortical surfaces of different subjects and each other, or between these surfaces and a geometric primitive such as the sphere. With such maps, intersubject studies

of cortical surface variation are possible using surface warping methods [3], [5]–[9]. Two-dimensional brain surface topology is frequently assumed for cortical flattening approaches, such as those described in [1], [2], [4], and [5]. The cortical mapping methods presented by Drury *et al.* and Fischl *et al.* require 2-D surfaces for flattening; such surfaces can be obtained by cutting a spherical surface [2], [5]. Topologically correct brain surfaces can also be used to constrain the location of neural current sources to the cerebral cortex when solving inverse problems in magnetoencephalography (MEG) and electroencephalography (EEG) [1], [10]. Applications such as these motivate the development of automated methods for generating cerebral cortex surface representations that have the appropriate topology.

Magnetic resonance (MR) imaging is the typical modality from which neuroanatomical surfaces are obtained because of its ability to provide high resolution, high contrast images of the soft tissues of the human brain. The task of identifying structures in MR imagery is typically approached from one of a few directions. Registration techniques can be used to deform a labeled atlas to a subject to identify neuroanatomy [11]–[13]. These methods typically perform well in subcortical structures such as the basal ganglia but have difficulty matching the cortex due to the variability found among different subjects. For this reason, Collins *et al.* incorporated a low-level tissue classification scheme with a deformable atlas in order to label regions of the cortex [13].

Another technique for identifying anatomical structures is the use of deformable surface methods, several of which have been surveyed by McInerney and Terzopoulos [14]. These methods can be implemented such that the topology of the final surface matches that of an initial template. However, this can be computationally intensive compared to low-level processing, and the deforming surface must be prevented from intersecting itself. Also, these methods do not always deform well into cortical structures due to other attractors in the image. Accurate segmentation using deformable surfaces often depends upon a reasonably accurate initialization [15].

Low-level tissue classification approaches, in which individual voxels in the MR image are labeled according to tissue types, can provide high-resolution detail of the cortical surface and are often used to initialize active contour models. These techniques are susceptible to inhomogeneity artifacts, partial voluming, and other noise, which will result in mislabeled voxels. Additionally, the low resolution of the scanning system, as compared to the true fine scale of the neuroanatomy, will often obscure the boundaries between neighboring gyri. For these reasons, tessellations made from low-level labelings of the cerebral cortex typically exhibit numerous topological

Manuscript received June 2, 2000; revised August 12, 2001. This work supported by the National Institute of Mental Health under Grant R01-MH53213, by the National Center for Research Resources under Grant P41-RR13642, and by the National Foundation for Functional Brain Imaging. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was W. Niessen. *Asterisk indicates corresponding author.*

*D. W. Shattuck is with the Signal and Image Processing Institute, Department of Electrical Engineering—Systems, University of Southern California, 3740 McClintock Ave., Suite 404, Los Angeles, CA 90089-2564 USA (e-mail: shattuck@sipi.usc.edu).

R. M. Leahy is with the Signal and Image Processing Institute, Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089-2564 USA.

Publisher Item Identifier S 0278-0062(01)10001-7.

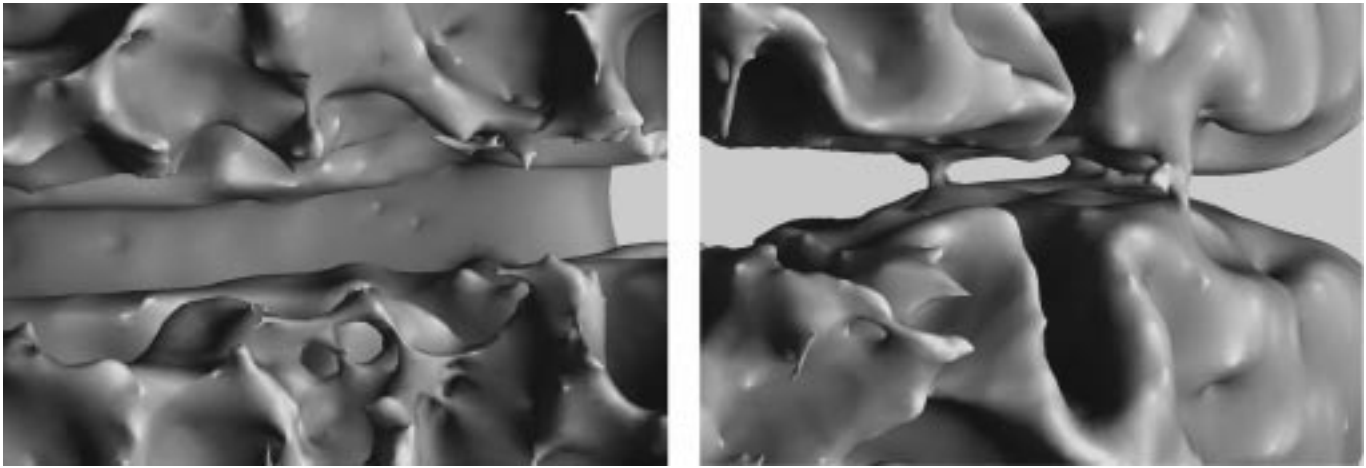


Fig. 1. Topological errors in a cortical surface. Two close-up views of the grey matter/white matter cerebral boundary surface generated from a low-level tissue classification of an MR image. Several topological handles are clearly visible, including bridges between the cerebral hemispheres.

defects in the form of small handles or tunnels on the surface or even false connections between the hemispheres, as seen in Fig. 1.

The method presented by Teo *et al.* provides for identification of potential topological defects which can then be corrected by hand; verification of the Euler characteristic on the volume ensures that the corrected volume will have the appropriate topology [16]. The cortical surface-based analysis methods presented by Dale, Fischl, and Sereno [4], [5] addressed this problem using hand editing for larger topological defects while ignoring smaller ones that do not impact their flattening methods. Identification of the topological errors was performed by viewing an inflating cortical surface, and correction was achieved by editing the surface or the data. Interactive correction of surface meshes, or the volumes from which they are generated, can be difficult and time consuming, particularly for errors involving only a few voxels. More recently, Fischl *et al.* presented a method to perform automated topological surgery on the initial cortical surface [17]. This method inflates the cortex to a sphere and identifies overlapping triangles in the spherical surface. These areas occur due to topological handles, and are retessellated to correct the topological defects.

The cortical identification method presented by Xu *et al.* also addresses the issue of automatically correcting surface topology [18], [19]. Their method makes use of fuzzy classification to produce an initial isocontour representing the boundary between white matter and grey matter. If this surface is not homeomorphic to a sphere the fuzzy membership set is median-filtered spatially and the isocontour surface is recomputed. This procedure is iterated until a surface with the appropriate topology is found. The topologically correct isocontour surface is then used as an initial condition for an active contour model that finds the medial grey matter cortical surface using a gradient vector flow method [20]. While this approach has been shown to produce topologically spherical surfaces, the use of median filtering on the membership set affects the entire volume and can smooth certain aspects of the cortex. A second drawback to this approach is that the use of a $3 \times 3 \times 3$ median filter does not guarantee a topologically spherical result. An example of this is a configuration of voxels with two-voxel-thick sheets connected

by a number of 2×2 bridges. The structure will not have spherical topology, but will be invariant under the median filter.

In this paper, we present a method designed to take a binary volumetric object with an arbitrary topology and automatically edit its membership such that a marching cubes [21] tessellation of the edited object will be homeomorphic to a sphere. The method identifies topological problems in the volume prior to tessellation and makes minimal changes in the volume. This method is suitable for use in cortical surface identification because of its ability to find and then break or fill the small topological defects that occur during classification. Most significantly, the method identifies specific locations within the volume where topological defects form and can be corrected with changes to the membership set that are local to the area of the proposed defect. In most cases, these changes are on the order of one or two voxels. In this way, the algorithm respects the geometry of the cerebral cortex when making its changes to the volume. We demonstrate the method on a simple test object and on several white matter volumes generated from MR brain images.

II. METHOD

Our method analyzes and corrects the topology of volumetric binary objects. We assume the object input to our algorithm is a single, wholly connected foreground object with no internal cavities. In this context, we mean connected in the six-neighbor sense, where two voxels are neighbors if and only if they share a common face. Because the object has no internal cavities, the background is a single connected region. For background connectivity, we use the D18 rule, where any two voxels are neighbors if and only if they share a common face or edge. Illustrations of connectivity rules are shown in Fig. 2.

A. Graph Based Topological Analysis of Volumetric Data

The topological genus of a surface represents the number of nonintersecting closed curves that can be inscribed on the surface without separating the set of surface points into disjoint sets. In more common terminology, the genus represents the number of holes in the object. For simple polyhedra, which are

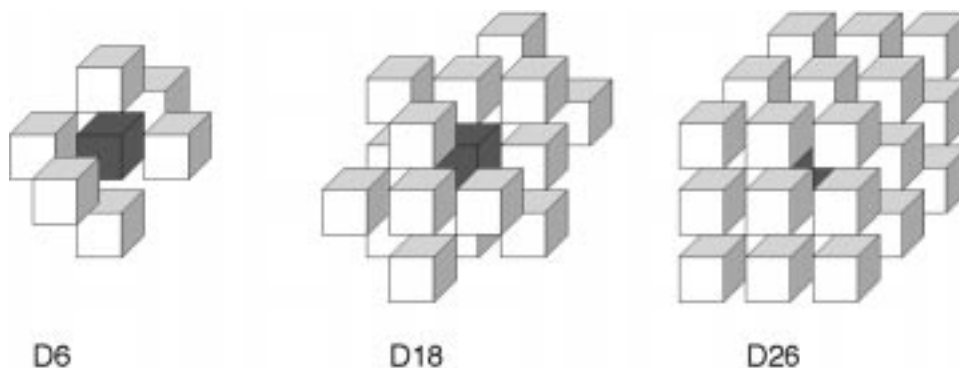


Fig. 2. Examples of connectivity rules. A (shaded) voxel and its D6, D18, and D26 neighbors.

homeomorphic to a sphere, the genus is zero. The genus of an object can be determined from its Euler characteristic, which can be computed from a tessellation of the object according to the Euler–Poincaré formula

$$\chi = V - E + F = 2 - 2g \quad (1)$$

where

- χ Euler characteristic;
- V number of vertices;
- E number of edges;
- F number of faces;
- g topological genus [22].

For an object with the topology of a sphere, $\chi = 2$. The genus may also be computed as a summation of the object’s local Euler characteristics [16]. However, these computations provide no spatial information regarding the position or size of the topological handles present in the volume since, from a topological viewpoint, such information is irrelevant. For this reason, we analyze the topology of the object while also analyzing its geometry. We form two weighted graphs, one for foreground voxels and one for background voxels, that contain information about the volume. Based on a conjecture described below, this allows us to determine if the foreground object is topologically equivalent to a sphere by counting the number of cycles in the graphs. The cycles also provide information about the size and location of handles or holes.

1) *Foreground Connectivity Graph*: We create a foreground connectivity graph that captures important information about the structure of the binary image foreground. We examine the object along a selected cardinal axis, identifying the connected components within each slice. Each in-slice connected component, which we call a node, corresponds to a vertex in the graph.

We next analyze how each of the nodes is connected to the nodes in the slices above and below it. We define a connection between two nodes as a contiguous collection of voxels in one node that share a face with a contiguous collection of voxels in a node in an adjacent slice. Each connection in the object is represented by an edge in the graph. Importantly, two nodes may be connected in more than one place. In such a case, they will have multiple edges connecting their vertices in the foreground connectivity graph. Such an occurrence is the simplest form of a topological handle existing in the object.

The strength of the connection between two nodes in the object is represented in the graph by a weight associated with the edge. In this paper, we use a weight function that measures the number of voxels in a connection between nodes in adjacent slices. In this way, nodes that are very strongly connected will have large weights, while nodes that touch at only a few voxels will have small weights. A simple example of a foreground connectivity graph, computed for a toroidal object, is shown in Fig. 3. The cycle in this graph corresponds to the handle (or hole) in the object.

We base our connectivity rules for the connections in the foreground graph on the D6 rule, with a slight modification that makes our method appropriate for the marching cubes algorithm. We start by taking the intersection of two adjacent slices and consider each D6 connected component within this intersection to be a single connection between nodes. However, as shown in Fig. 4, a $2 \times 2 \times 2$ cube of foreground voxels with opposite corners removed will produce two distinct connections using D6 connectivity, implying a topological hole according to our algorithm. This is contrary to a marching cubes tessellation of this object, which will not have a hole. Our algorithm examines connected regions that have a voxel sharing an edge with a voxel in another connected region to see if voxels in that region are in the configuration shown in Fig. 4. If so, the two connections are merged. This reduces the number of corrections made to the object, improving its fidelity to the initial classification [23].

2) *Background Connectivity Graph*: Though cycles in the foreground connectivity graph represent the possible presence of a topological handle or hole, the foreground graph itself is not sufficient for testing the topology of the object. If the torus of Fig. 3 is rotated 90° such that its hole is vertical relative to the slicing axis, also shown in the figure, then it will not generate a cycle in its foreground connectivity graph. Because of such discrepancies, we also form a background connectivity graph. Once again, we find each of the connected background components, which we describe as background nodes, in each slice of the object. These are represented by vertices in the background graph. Here, we use D18 connectivity rules to identify the connected components. This means that within a slice, connected components will share a face or edge. Connections between adjacent nodes and the strengths of these connections are then found, and represented in the background connectivity

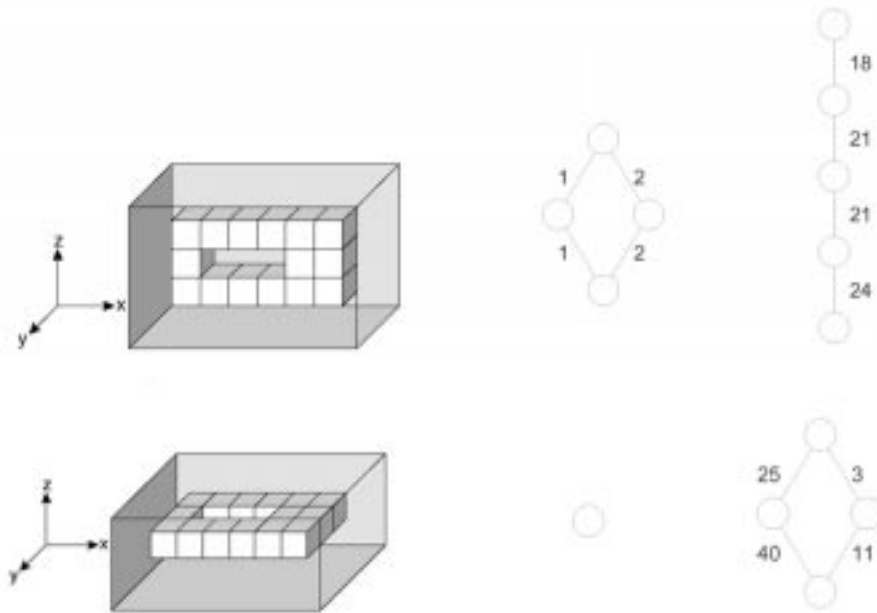


Fig. 3. Analysis examples. (From left: Object, foreground graph, and background graph) (top) An $8 \times 3 \times 5$ binary volume containing a torus, and its connectivity graphs. Analysis is performed along the z axis. The hole in the object appears as a cycle in the foreground graph. (bottom) The same object, rotated 90° , and its connectivity graphs. The hole in the object now appears as a cycle in the background graph. The foreground appears as a single node. Analysis is performed along the z axis.

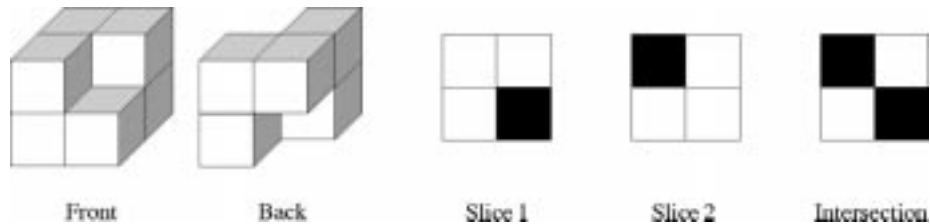


Fig. 4. Special connectivity rule. This example shows a $2 \times 2 \times 2$ configuration of voxels consisting of six foreground voxels such that two opposite corners of the cube are background. Taking the intersection of the two slices shows the D6 connections between the two slices. The intersection in this case suggests two connections between the slices and the presence of a topological handle. However, no such handle is present in a marching cubes tessellation of this object. Representing these two apparent connections as a single connection in the foreground connectivity graph makes our algorithm more appropriate for use with the marching cubes algorithm.

graph as weighted edges. These connections are found by taking the intersection of pairs of connected components from adjacent slices after dilation of one of these components with a cross operator. This identifies connections formed by components sharing an edge, but not those that only share a corner. An edge is added to the background graph for each connected component in this intersection, with the edge weight defined by the component's size. It should be noted that for some cases the size of connection will be dependent on which node is dilated prior to taking the intersection; we intend to address this in future implementations of our method. Background connectivity graphs are also shown in Fig. 3.

3) *Spherical Homeomorphism Conjecture*: We conjecture that if the foreground and background connectivity graphs for an object without interior cavities are both trees (graphs with no cycles), then the object's surface is homeomorphic to a sphere. This forms the basis for our correction strategy. Though we have not proved our conjecture, it has been true for each simulated and real object we have analyzed.

B. Automated Topological Editing of Volumetric Data

Low-level segmentation results will not typically exhibit spherical topology. In this section, we describe methods by which the topology of a volumetric object may be altered so that its surface will be homeomorphic to a sphere. It is our conjecture that if an object does not have a topologically spherical surface, then its connectivity graphs will have cycles. Thus, our strategy for correcting an object's topology is to alter the object's membership set in such a way that it will generate foreground and background connectivity graphs that have no cycles. We first identify a pair of desired graphs using Kruskal's algorithm [24] to produce two maximal spanning trees. Since the weighting of our graphs represents the strength of connection between nodes in adjacent slices, the maximal spanning tree algorithm will remove the lightest possible set of edges from the graphs, corresponding to the weakest connections, in order to create trees. The task of our editing procedure is to break the links in the actual object that represent the edges deleted from the graphs.

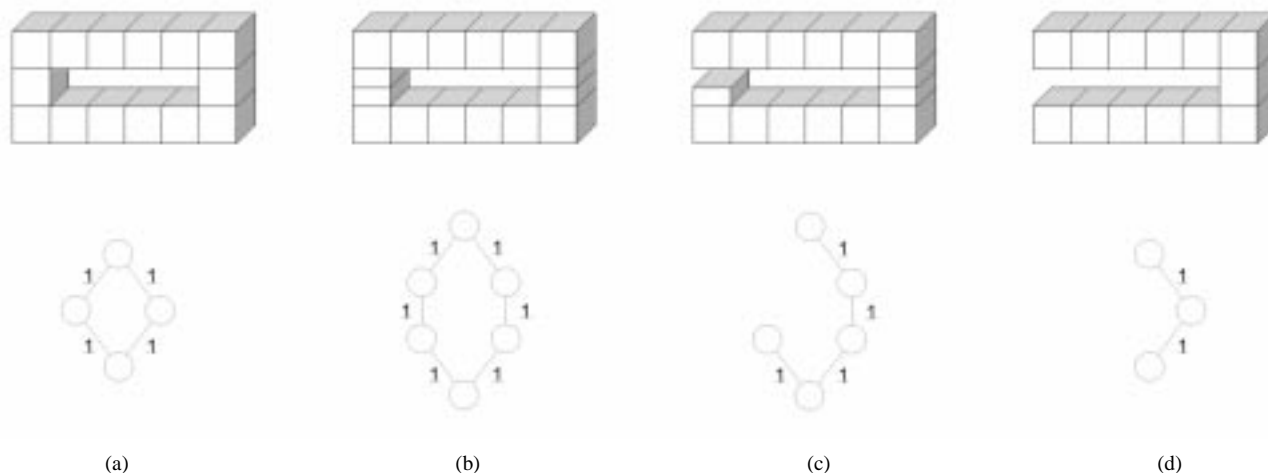


Fig. 5. Correcting object topology. We edit an object to remove a handle. (a) The original object and its foreground connectivity graph. (b) We subsample the object in a slice adjacent to the connection we are breaking. The foreground connectivity graph is expanded. (c) We can then remove a node in the graph and the object; this breaks the cycle in the graph and causes the object to have a genus of 0. (d) In this simple case, we could have simply removed a node from the graph and object.

1) *In-Place Correction*: It is important to recognize that each edge connecting two nodes in the graphs corresponds to connections between voxels and not voxels themselves. Still, both nodes contain subsets of voxels that form the edge. If we naively remove either set of voxels from the nodes, we may break links to other nodes in the object. This could result in separating the object into multiple pieces. For this reason, we have devised three tests for sets of voxels before we alter their membership in the object. These tests are described below in order of increasing computational complexity. A simple correction example appears as Fig. 5. When removing voxels from the foreground nodes, we are removing them from the object. Conversely, when we remove voxels from the background nodes, we are adding them to the object.

Our first test is based on the graph itself. Each edge to be removed connects two nodes that will remain in the graph after correction and are, therefore, part of the desired maximal spanning tree. If either of these nodes is a leaf on the tree, i.e., if it is only connected to one other node on the tree, then we can remove the voxels from this node and sever the link. The remaining edges in the graph will be unaffected. It may be the case that other voxels in this node are separated from the main object; however, other nodes are unaffected. It may also be the case that removing voxels from the foreground or background nodes will produce cycles in the opposite graph; however, this can be corrected in subsequent operations. In practice, we have not observed situations where cycling between solutions occurs.

The second test we apply is to examine the connectivity in the area of the two possible corrections. This is best explained with an example. Assume we want to remove a set of foreground voxels S_1 from a node in slice n , and the S_1 voxels form an undesired connection with the set of voxels S_2 in slice $n + 1$. If the neighbors of S_1 in slice $n - 1$ are also foreground voxels, then these voxels will remain connected after we remove the S_1 voxels. Furthermore, they will remain connected to the rest of the object because they would otherwise have been only connected to S_1 , and the edge formed between S_1 and S_2 would

have been part of the maximal spanning tree. Thus, the S_1 voxels can be removed without altering any other nodes in the object. If this test is not passed, we do the same test with S_2 and slice $n + 2$. A similar test is used for background voxels.

Our third test is to simply remove the voxels that form a link from either node and examine the object or background to see if our correction caused it to become a disjoint set. If so, we undo the correction. Otherwise, we accept the correction and proceed to the next problem.

2) *Subsampling Method*: In some cases, we may be unable to find a suitable way of breaking a link in the tree by only removing or adding voxels to the object. Instead, we can subsample the volume at the location of the topological defect. More specifically, we can insert a slice between the two slices where the link occurs; this slice will be a duplicate of either slice. We have effectively cut one of the slices in half.

Splitting the slice in two introduces duplicate nodes in the graph for the original slice. Assume we duplicate slice n with slice n' , and now the slices appear in the volume in the order $\{n, n', n + 1\}$. Given a node N_b in n' , N_b is linked only to the node it duplicates in slice n , say N_a , and the nodes in $n + 1$ to which N_a was connected. We are free to remove the voxels that form an undesired link between N_b and the nodes in slice $n + 1$. Any other voxels in N_b will remain connected to N_a and, thus, any other connections between N_b and nodes in slice $n + 1$ will remain connected to the object. A simple example is shown in Fig. 5.

We keep an index to these slices as being only half of their original height so that we can compensate for this in tessellation or other procedures. Clearly, the in-place correction method is preferred as subsampling can greatly increase the size of the volume and subsequent tessellations. However, having this method as a last resort ensures that we can always generate an object that produces appropriate graphs. In practice, we have not encountered a brain volume that required the subsampling correction method. The subsampling method is described in more detail in [25].

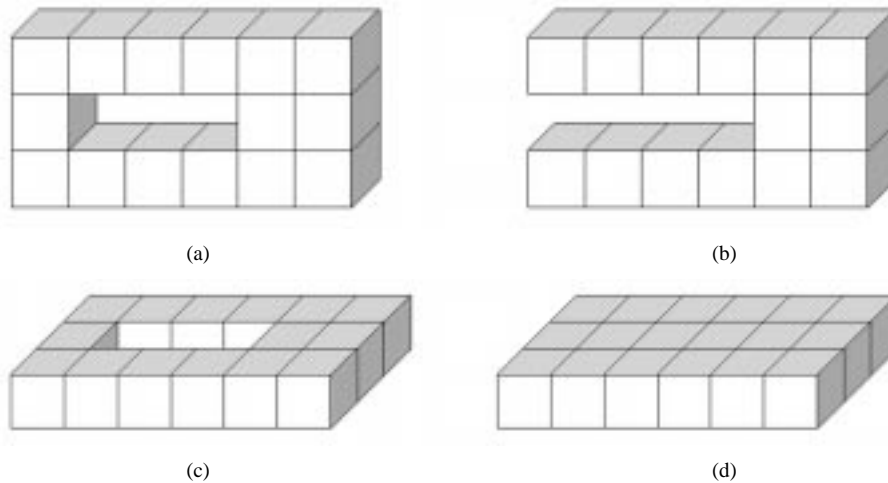


Fig. 6. Axis dependence of corrections. The corrections applied to an object vary depending on the axis along which the volume is analyzed. (a) and (b) The object from Fig. 3 is corrected by removing a single voxel. (c) and (d) The same object analyzed along a different axis is corrected by adding three voxels.

C. Iterative Multiaxis Correction

The nature of the changes described in the previous section depends on the axis along which the object is analyzed. This is shown in Fig. 6, where a topological handle is fixed by either removing one voxel or adding three. To ensure that we make the smallest changes possible to the object, we iteratively apply the corrections along each axis. A threshold is used, and only corrections less than or equal to the threshold are performed. This multiaxis approach dramatically reduces the number of voxels that are added to or removed from the object membership set [25], [23].

D. The Topological Constraint Algorithm

The topological constraint algorithm is summarized below. The algorithm assumes as input a single D6 connected object in a single D18 background; i.e., the object has no interior background cavities.

- 1) Set the threshold value to one.
- 2) Starting with the z axis, compute a foreground connectivity graph $G = \{N, E\}$.
- 3) Compute a maximal spanning tree $T = \{N, \tilde{E}\}$ from $G = \{N, E\}$. The set $E_d = E - \tilde{E}$ represents the set of edges to be deleted from the tree.
- 4) Find the set of edges $E_t \subset E_d$ that have weights less than or equal to the threshold.
- 5) For each edge $e \in E_t$ where either node is connected to only one other node in T , remove the voxels that form the link from the slice where the node is singly connected.
- 6) For each edge remaining in E_t , test the connectivity in the slices above and below to see if the linking voxels can be removed from the node; if so, remove them.
- 7) For each edge remaining in E_t , remove the linking voxels and test to see if the correction is acceptable. If not, restore the voxels to their previous state.
- 8) Repeat steps 2–7 for the x - and y axes.
- 9) Repeat steps 2–8 for the background components in the object.
- 10) Repeat steps 2–9 until no further changes can be made at this threshold.

- 11) Iterate steps 2–10 with an increasing threshold until no further corrections can be made.
- 12) If cycles still exist, conclude by inserting duplicate slices where necessary and remove undesired linking voxels in the duplicate slices.

III. RESULTS

We implemented the topological constraint algorithm in C++ and successfully applied it to volumetric objects. We present the results of the algorithm on a small test object and demonstrate the graph formation and correction process. We then present the results of using the algorithm on several cerebral cortex masks generated from MR images of several subjects.

A. Simple Test Objects

Fig. 7 shows two renderings of a small hand-created object composed of 65 connected voxels. The voxels are contained within four slices of a $20 \times 20 \times 6$ volume. Computation of the Euler–Poincaré formula on the marching cubes [21] tessellation shows that the object has a genus of five. The handles or holes of the object are clearly visible; we see three holes aligned with the vertical axis and two aligned horizontally. The interslice connectivity graphs for this object are also shown in Fig. 7. Each of the three vertical holes are captured as cycles in the foreground graph, while the two horizontal holes are captured in the background graph.

For illustrative purposes, we apply the algorithm along a single axis only. The foreground pass identifies the three foreground cycles, creates the desired maximal spanning tree, and removes three voxels to produce an object with genus 2. These graphs and the corrected object are shown in Fig. 8. The background analysis is then performed—two cycles are identified, a maximal spanning tree is created, and three voxels are added to the object to break the cycles. These graphs and the resulting object are shown in Fig. 9. This object has a genus of 0, as verified by the Euler–Poincaré formula. In this case, all corrections were made using the in-place correction approach. The connectivity graphs of this object are two trees, in accordance with our conjecture. A total of five voxels were

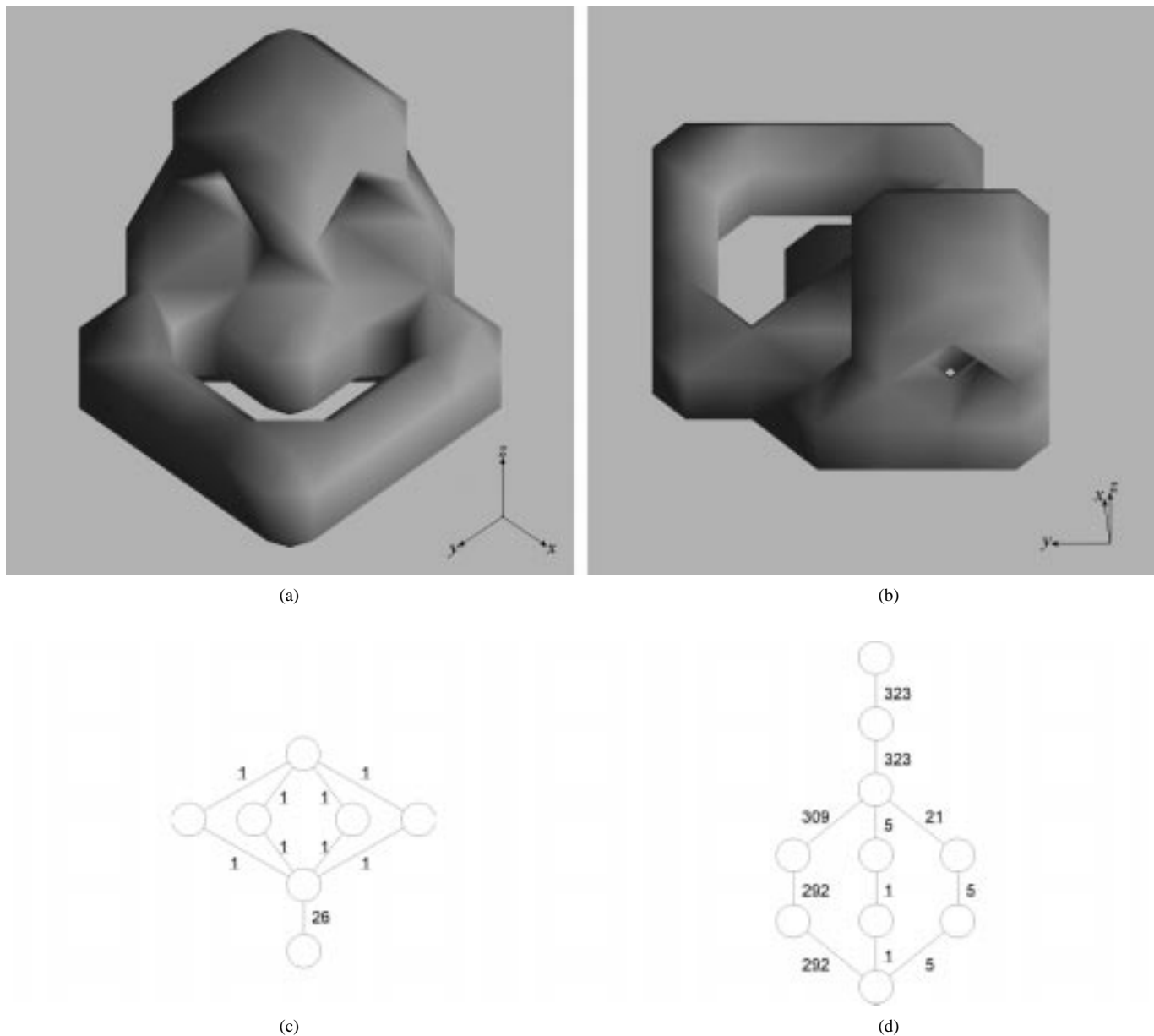


Fig. 7. Test object. (a) and (b) Renderings of a simple object used to test the topological constraint algorithm. (c) Connectivity graph for foreground. (d) Connectivity graph for background.

changed to constrain the topology of this object. While these changes appear very significant to the shape of the object, it should be noted that the object's structure is of the same scale as its topological defects.

B. Human MR Data Sets

The method described in this paper can be applied to arbitrary three-dimensional binary volumetric objects. Our specific application is to constrain an object representing the interior volume of the cerebral cortex such that its tessellation is topologically equivalent to a sphere. This problem provides an excellent test for the algorithm, as segmentations based on low-level tissue classification typically produce brain surfaces with numerous tiny handles due to classification errors. The use of the topological constraint algorithm may be justified by the small size of

these defects since it will choose the smallest regions it can in which to make its corrections.

1) *Preprocessing*: We preprocessed six T1-weighted MR volumes for this study using a sequence of low-level steps described in [26]. We first stripped skull and scalp from the volume using a combination of edge-enhancing anisotropic diffusion filtering, Marr–Hildreth edge detection, and mathematical morphology. We then compensated for image nonuniformity using a parametric tissue intensity model that adapts a tricubic B-spline gain field to match local tissue properties to the global properties of the image. Finally, the voxels within the brain volume were labeled as white matter, grey matter, cerebrospinal fluid (CSF), and partial volume combinations of these. The classification step used the same parametric model as the inhomogeneity compensation step, but in this case the model was formulated into a Bayesian classifier

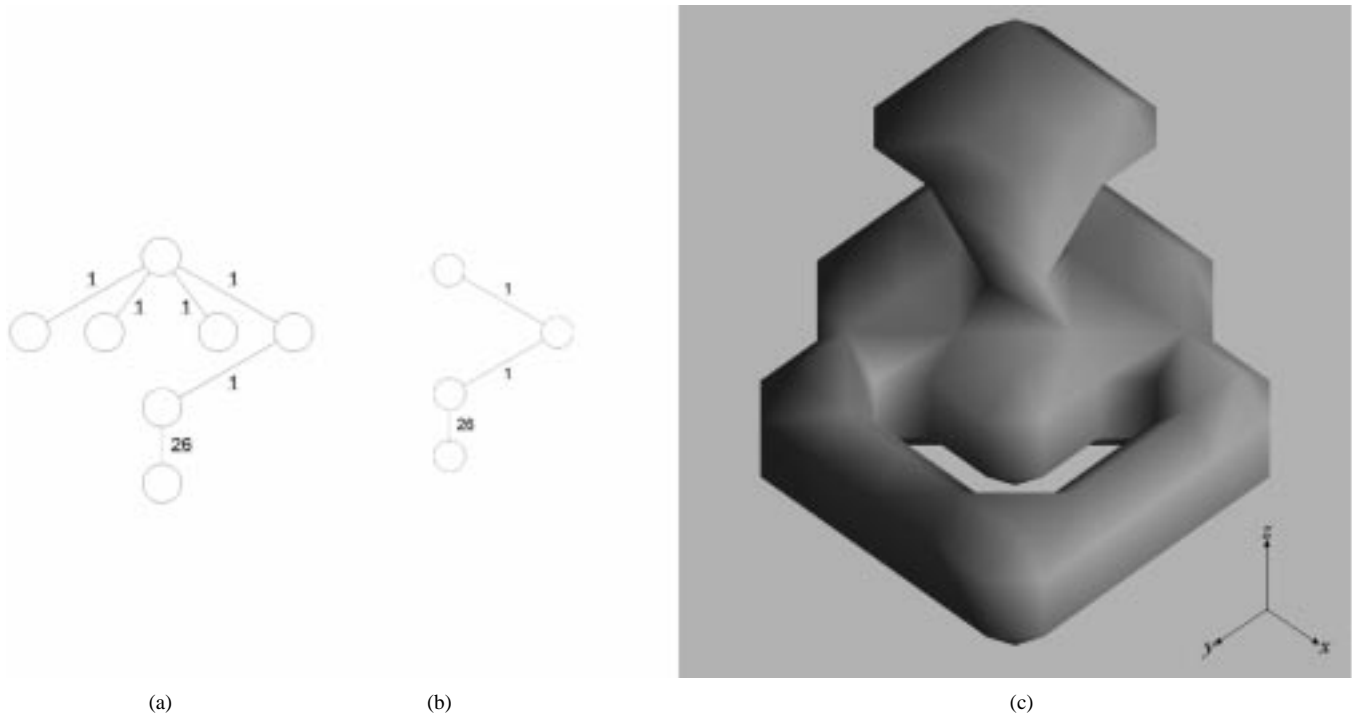


Fig. 8. Foreground correction. (a) The maximal spanning tree identified by the algorithm. (b) Connectivity graph of the object after foreground correction. (c) Rendering of the edited object.

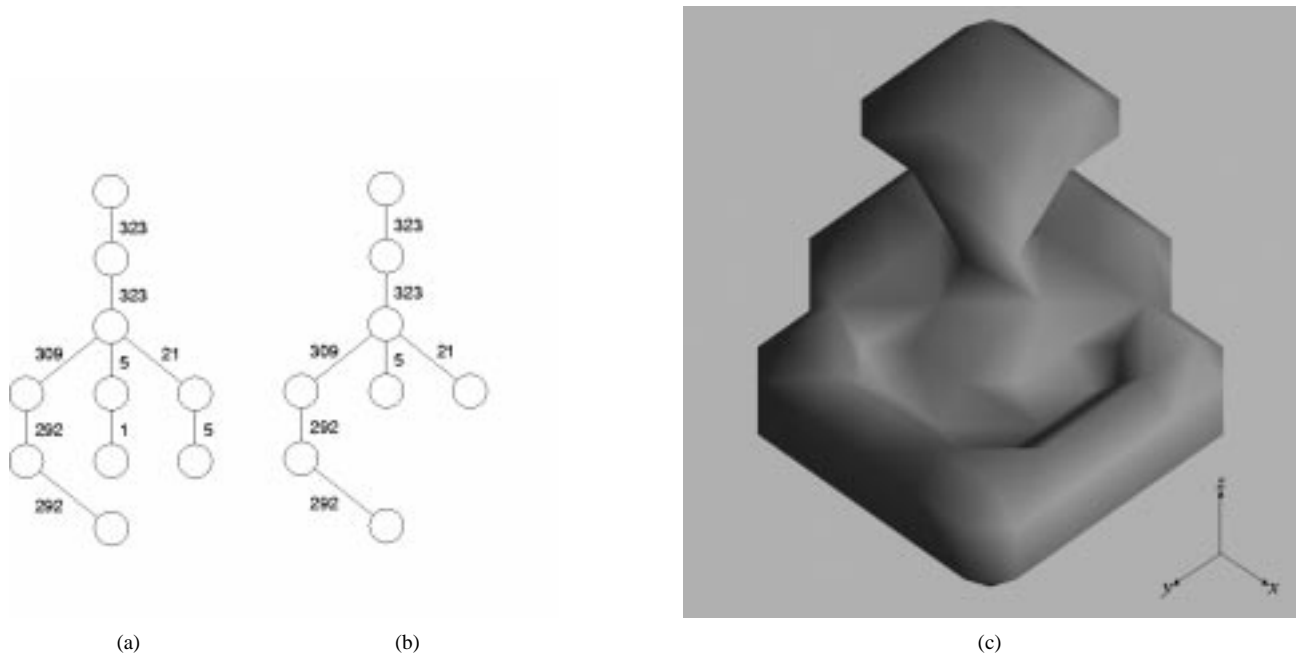


Fig. 9. Background correction. (a) The maximal spanning tree identified by the algorithm for background connectivity. (b) Connectivity graph of the object after background correction. (c) The corrected object with genus 0.

with a Gibbs spatial prior model to encourage piecewise contiguous labeled regions.

We used the output of the classification step to produce the initial white matter volume. The largest connected component composed of voxels containing white matter was selected, and additional processing was performed to compensate for certain segmentation irregularities. Specifically, in the area of the subcortical nuclei, interior grey matter often joined with sulcal CSF as a result of the low resolution of the acquisition system rela-

tive to the fine anatomical detail in the image. Thus, the initial volume typically included the surfaces of the ventricles. To account for this, we searched for the lateral ventricles by analyzing the CSF voxels in the volume. If found, the ventricles were then included in the inner cerebral white matter volume; otherwise they were filled manually. Hand editing was also required to ensure that the complete subcortical nuclei structures were also included inside the initial white matter volume. It should be noted that if structures such as the basal ganglia are left unfilled

TABLE I

CHANGES MADE TO CORRECT TOPOLOGY. THE GENUS OF EACH OBJECT WAS REDUCED BY OUR ALGORITHM TO ZERO WITHOUT THE NEED FOR SUBSAMPLING. IN EACH CASE, THE NUMBER OF VOXELS CHANGED WAS VERY SMALL COMPARED TO THE SIZE OF THE BRAIN VOLUME, AND FEW LARGE CHANGES WERE MADE

	initial brain size (voxels)	initial genus	final genus	number of changes	voxels changed	pct. changed	largest change	number of changes > 10
1	411952	207	0	217	464	0.11%	38	5
2	218055	129	0	136	449	0.21%	54	8
3	641153	201	0	217	436	0.07%	40	5
4	444990	237	0	368	712	0.16%	41	6
5	648132	233	0	164	360	0.06%	35	4
6	455896	109	0	111	257	0.06%	37	3

TABLE II

SIZE AND NUMBER OF CHANGES MADE TO EACH WM VOLUME. MOST CHANGES MADE ARE ONE OR TWO VOXELS IN SIZE; A FEW LARGE CHANGES OF SIZE LARGER THAN 10 ARE REQUIRED FOR EACH VOLUME

volume	Total	Number of changes made per size										
		1	2	3	4	5	6	7	8	9	10	>10
1	217	160	24	16	4	1	0	4	2	1	0	5
2	136	95	16	6	4	4	2	0	0	1	0	8
3	217	156	28	12	4	4	2	2	0	3	1	5
4	368	267	57	18	10	5	3	0	1	0	1	6
5	164	119	23	10	3	2	2	0	0	1	0	4
6	111	71	24	5	5	1	1	0	0	1	0	3

so that bridges in the white matter exist, these structures will be severed or filled by the topology correction algorithm. We cut the volume at the brain stem to remove the cerebellum. The resulting volume was then processed with the topological constraint algorithm to force its tessellation to be homeomorphic to a sphere.

2) *Topological Correction Results:* We applied the topological constraint algorithm to six T1-weighted MR volumes after preprocessing using the procedure described above. All processing was performed using the BrainSuite tool [27]; processing time for each volume was between 5 and 10 min on an Intel Pentium III Xeon 933-MHz CPU. It should be noted that aspects of the implementation of our approach, such as the frequent recalculation of the graphs, can be optimized, providing for substantial decreases in computational cost.

Each white matter volume was tessellated after topological correction and the genus was verified by the Euler–Poincaré formula. In every case, the method produced a final volume with a tessellation having an Euler characteristic of two, corresponding to a genus of zero and homeomorphism to a sphere. Importantly, the algorithm was able to correct all volumes without subsampling and with very few changes to the membership set.

Table I shows the initial genus, number of changes, and largest single change made to each volume. Further details of the changes made are provided in Table II, which shows the number of changes made for each size. Fig. 12 shows a histogram of the fraction of changes made of each given size, averaged over the six volumes. Our algorithm changed less than 0.21% of the voxels in any volume, and 90% of the changes were made by altering three or fewer voxels. Most changes made were less than ten voxels in size. The larger changes are unlikely to closely follow the anatomical contours of the image since they are restricted to be within a single plane. However, the algorithm can be run with a limit on the largest change made to the volume. The larger changes can then be made manually.



Fig. 10. Slice from a topology corrected image. White voxels are those voxels common to the precorrected and corrected image. Black voxels have been changed by the algorithm to correct the topology. The changes made by the algorithm tend to be only a few voxels per topological defect.

The small nature of the changes made is highlighted by Fig. 10, which shows the corrections made to a single slice in one of the volumes. Fig. 11 shows renderings of marching cubes tessellations of one of the volumes before and after correction. The cortical surfaces are identical except in regions where topological defects have been broken or filled.

To analyze the axis dependency of our corrections, we applied the algorithm to each volume after transposing the x and z axes, and the y and z axes. Ideally, the set of voxels changed would be independent of orientation. In most cases, at least 70% of the voxels changed were independent of orientation. However, there were volumes in which only 25% of the voxels changed were independent of orientation. This could be addressed in future work by combining the analysis of all three axes.

A more important problem with our approach is that each edit made to the object is restricted to changing voxels within a single slice of data. Anatomical boundaries in the volume are not likely to line up along these boundaries, which means that as the size of the correction gets larger it is more likely to deviate from the anatomical boundary. This can result in notches

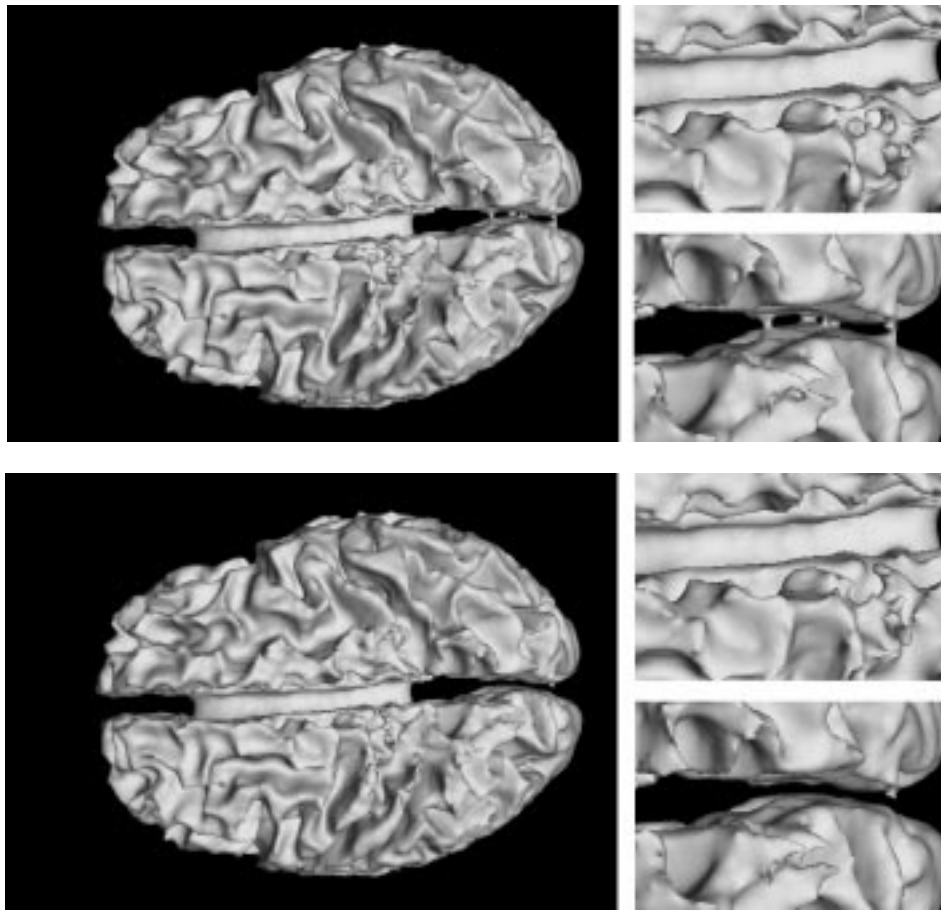


Fig. 11. Topological constraint results. The top rendering shows a marching cubes tessellation of a white matter surface following a smoothing to reduce staircase artifacts due to the binary nature of the data. Several topological defects are clearly visible in the image. The lower rendering shows the same volume after application of the topological correction algorithm, after tessellation and smoothing by the same process. All topological handles have been filled or severed but the global properties of the figure remain nearly identical. Inset are close-ups of corrected topological defects.

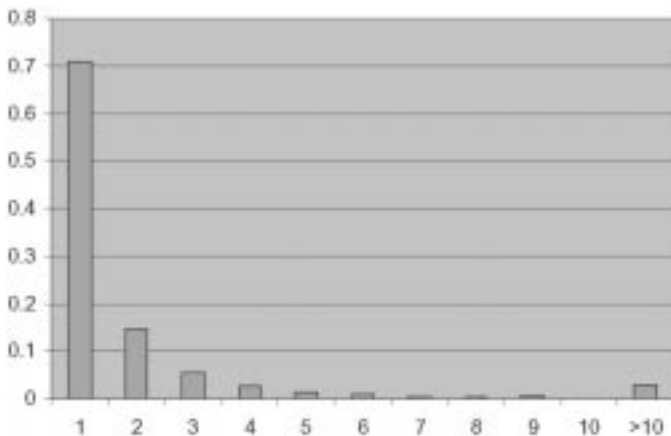


Fig. 12. Histogram of average fraction of changes made per size. Averaged over the six brains in Table I.

in the final surface, though the use of the multi-axis iterative approach significantly reduces this. In future work, we intend to explore approaches that will allow larger corrections to follow more natural contours. One possible method would involve removing only a portion of each defect during an iteration, where the portion removed will be selected based on the intensity of the MR data. For our cortical surface identification algorithm,

this would allow us to remove voxels forming a topological defect that are less likely to be white matter.

IV. CONCLUSION

In this paper, we have presented a new method that analyzes and alters the topology of a binary volumetric object so that its tessellation will be homeomorphic to a sphere. The method is appropriate for forcing the topology of cerebral white matter volumes to be spherical. It operates on the voxel membership set of the volume, and makes minimal changes to force it to have the topology of a sphere. The method has correctly altered the topology of every volume tested.

This method provides a completely automated way to quickly correct the topology of cortical volume segmentations. When used in combination with largely automated segmentation procedures, the burden on human operators to produce topologically correct cortical surface models is significantly reduced. This increases the scale on which surface-based intersubject population studies can realistically be performed.

ACKNOWLEDGMENT

The authors gratefully acknowledge the valuable contributions made by the reviewers of this paper.

REFERENCES

- [1] A. M. Dale and M. I. Sereno, "Improved localization of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: A linear approach," *J. Cogn. Neurosci.*, vol. 5, no. 2, pp. 162–176, 1993.
- [2] H. A. Drury, D. C. Van Essen, C. H. Anderson, C. W. Lee, T. A. Coogan, and J. W. Lewis, "Computerized mappings of the cerebral cortex: A multiresolution flattening method and a surface-based coordinate system," *J. Cogn. Neurosci.*, vol. 8, no. 1, pp. 1–28, 1996.
- [3] P. M. Thompson, D. MacDonald, M. S. Mega, C. J. Holmes, A. C. Evans, and A. W. Toga, "Detection and mapping of abnormal brain structure with a probabilistic atlas of cortical surfaces," *J. Comput. Assist. Tomogr.*, vol. 21, no. 4, pp. 567–581, 1998.
- [4] A. M. Dale, B. Fischl, and M. I. Sereno, "Cortical surface-based analysis I: Segmentation and surface reconstruction," *NeuroImage*, vol. 9, pp. 179–194, 1999.
- [5] B. Fischl, A. M. Dale, and M. I. Sereno, "Cortical surface-based analysis II: Inflation, flattening, and a surface-based coordinate system," *NeuroImage*, vol. 9, pp. 195–207, 1999.
- [6] M. Vaillant and C. Davatzikos, "Hierarchical matching of cortical features for deformable brain image registration," in *Proc. 16th Int. Conf. Information Processing in Medical Imaging*, A. Kuba, M. Sámal, and A. Todd-Pokropek, Eds., June 1999, pp. 182–195.
- [7] P. Thompson, R. Woods, M. S. Mega, and A. W. Toga, "Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain," *Human Brain Mapping*, vol. 9, no. 2, pp. 81–92, 2000.
- [8] P. M. Thompson and A. W. Toga, "A surface-based technique for warping three-dimensional images of the brain," *IEEE Trans. Med. Imag.*, vol. 15, pp. 1–16, Aug. 1996.
- [9] H. A. Drury, D. C. Van Essen, M. Corbetta, and A. Z. Snyder, "Surface-based analysis of the human cerebral cortex," in *Brain Warping*. New York: Academic, 1999, pp. 337–363.
- [10] J. W. Phillips, R. M. Leahy, J. C. Mosher, and B. Timsari, "Imaging neural activity using MEG and EEG," *IEEE Eng. Med. Biol., Mag.*, vol. 16, no. 3, pp. 34–42, 1997.
- [11] M. Miller, Y. Amit, G. E. Christensen, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," in *Proc. Nat. Acad. Sci.*, vol. 90, 1993, pp. 11 944–11 948.
- [12] G. E. Christensen, "Consistent linear-elastic transformations for image matching," in *Proc. 16th Int. Conf. Information Processing in Medical Imaging*, A. Kuba, M. Sámal, and A. Todd-Pokropek, Eds., 1999, pp. 224–237.
- [13] D. L. Collins, A. P. Zijdenbos, W. F. Baaré, and A. C. Evans, "ANIMAL+INSECT: Improved cortical structure segmentation," in *Proc. 16th Int. Conf. Information Processing in Medical Imaging*, A. Kuba, M. Sámal, and A. Todd-Pokropek, Eds., 1999, pp. 210–223.
- [14] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.
- [15] A. Kelemen, G. Székely, and G. Gerig, "Elastic model-based segmentation of 3-D neuroradiological data sets," *IEEE Trans. Med. Imag.*, vol. 18, pp. 828–839, Oct. 1999.
- [16] P. C. Teo, G. Sapiro, and B. A. Wandell, "Creating connected representations of cortical gray matter for functional MRI visualization," *IEEE Trans. Med. Imag.*, vol. 16, pp. 852–863, Dec. 1997.
- [17] B. Fischl, A. Liu, and A. M. Dale, "Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex," *IEEE Trans. Med. Imag.*, vol. 20, pp. 70–80, Jan. 2001.
- [18] C. Xu, D. L. Pham, M. E. Rettmann, D. N. Yu, and J. Prince, "Reconstruction of the central layer of the human cerebral cortex from MR images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 467–480, June 1999.
- [19] D. Pham and J. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 737–752, Sept. 1999.
- [20] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, pp. 359–369, Mar. 1998.
- [21] W. Lorensen and E. Harvey, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM Comput. Graph.*, vol. 21, no. 24, pp. 163–169, 1987.
- [22] E. W. Weisstein, *The CRC Concise Encyclopedia of Mathematics*. Boca Raton, FL: CRC, 1998.
- [23] D. W. Shattuck and R. M. Leahy, "Topologically constrained cortical surfaces from MRI," *Proc. SPIE*, vol. 3979, Medical Imaging 2000: Image Processing, pp. 747–758, Feb. 2000.
- [24] R. Sedgewick, *Algorithms in C*. Reading, MA: Addison-Wesley, 1990, vol. Medical Imaging 1999: Image Processing.
- [25] D. W. Shattuck and R. M. Leahy, "Topological refinement of volumetric data," *Proc. SPIE*, vol. 3661, Medical Imaging 1999: Image Processing, pp. 204–213, Feb. 1999.
- [26] D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, D. A. Rottenberg, and R. M. Leahy, "Magnetic resonance image tissue classification using a partial volume model," *NeuroImage*, vol. 13, pp. 856–876, May 2001.
- [27] D. Shattuck and R. Leahy, "Brainsuite: An automated cortical surface identification tool," in *Lecture Notes in Computer Science*, S. Delp, A. DiGoia, and B. Jaramaz, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. , 1935, Medical Image Computing and Computer-Assisted Intervention—MICCAI 2000, pp. 51–61.