Topology Correction in Brain Cortex Segmentation Using a Multiscale, Graph-Based Algorithm

Xiao Han, Chenyang Xu, *Member*, Ulisses Braga-Neto, *Member*, and Jerry L. Prince*, *Senior Member*

***Abstract*—Reconstructing an accurate and topologically correct representation of the cortical surface of the brain is an important objective in various neuroscience applications. Most cortical surface reconstruction methods either ignore topology or correct it using manual editing or methods that lead to inaccurate reconstructions. Shattuck and Leahy recently reported a fully automatic method that yields a topologically correct representation with little distortion of the underlying segmentation. We provide an alternate approach that has several advantages over their approach, including the use of arbitrary digital connectivities, a flexible morphology-based multiscale approach, and the option of foreground-only or background-only correction. A detailed analysis of the method's performance on 15 magnetic resonance brain images is provided.**

***Index Terms*—Digital topology, segmentation, brain imaging, graph analysis, cortical surface extraction.**

I. Introduction

RECONSTRUCTION of the brain cortical surface from magnetic resonance (MR) images is an important goal in medicine and neuroscience. Reconstructed cortical surfaces can be used to study brain geometry and quantify geometric variations across populations, in normal growth, and disease [1], [2]. They can also be used to assist in brain registration by sulcal matching [3], [4], and to visualize functional data acquired through other imaging techniques [5], [6].

Producing the correct topology is an important part of the cortical surface reconstruction process. A reconstructed cortical surface without a correct topology may lead to incorrect interpretations of local structural relationships and will prevent cortical unfolding [7]–[11]. Geometrically, the human cerebral cortex is a thin folded sheet of gray matter (GM) that lies inside the cerebrospinal fluid (CSF) and outside the white matter (WM) of the brain. If the opening at the brain stem is artificially closed, the surface of the cortex is topologically equivalent to a sphere. The major topological defect is the presence of one or more handles on the reconstructed surface; clearly, this is not the topology of the true cortex. As illustrated in Fig. 1, a handle is associated with a tunnel, which is actually a handle in the background if the roles of the interior (foreground) and the exterior (background) of the surface are interchanged.

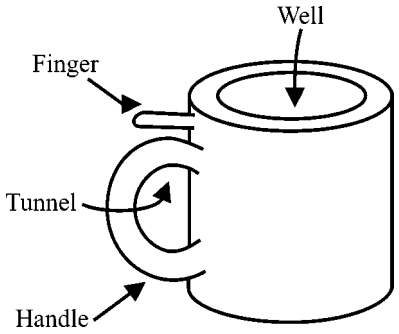

Fig. 1. Illustration of relevant topological features.

In recent years, there has been a considerable amount of work dedicated to the automatic extraction of cortical surfaces from MR images [8], [10]–[16]. Because of imaging noise, the partial volume effect, image intensity inhomogeneities, and the highly convoluted nature of the brain cortex itself, it is difficult to produce a representation that is both accurate and has the correct topology. Many methods either ignore this problem or rely on a manual postprocessing to correct the topology. For example, in [13], an anisotropic smoothing of the posterior probabilities of GM, WM, and CSF segmentation results is applied to obtain a binary segmentation of the WM. The resulting topology is manually checked and corrected if in error. The cortical surface analysis method proposed by Dale *et al.* [10], [14] also uses hand-editing to correct large topological defects, while ignoring smaller ones. They identify topological errors by viewing an inflated surface, and correct the error by editing the surface or the data.

There have been only a few methods that automatically produce topologically correct cortical segmentations [15]–[17]. The homotopically deformable region model proposed by Mangin *et al.* [17] is one of the best known. This method can also be adopted as a topology correction method. It starts with an initial region with the required topology, and then grows the region by adding points that will not change the original topology. Points that are allowed to be added or removed without changing the topology are known as *simple points* in the literature of digital topology [18]–[22]. The problem with this overall approach is that the result strongly depends on the order in which the points are grown from the initial volume. As an extreme example, it is possible for a volume that has only one small handle to be grown in such a way that the resulting

Manuscript received June 7, 2001; revised January 8, 2002. This work was supported in part by the National Science Foundation(NSF)/Engineering Research Center (ERC) under Grant CISST 9731748 and in part by the National Institutes of Health (NIH)/National Institute of Neurological Disorders and Stroke (NINDS) under Grant R01NS37747.The Associate Editor responsible for coordinating the review of this paper and recommending its publication was R. Leahy. *Asterisk indicates corresponding author.*

X. Han and U. Braga-Netos are with the Center for Imaging Science, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore MD 21218 USA.

C. Xu is with the Imaging and Visualization Department, Siemens Corporate Research, Inc., Princeton, NJ 08540 USA.

*J. L. Prince is with the Center for Imaging Science, Department of Electrical and Computer Engineering, Johns Hopkins University, 307 Barton Hall, 3400 North Charles Street, Baltimore MD 21218 USA (e-mail: jprince@jhu.edu).

Publisher Item Identifier S 0278-0062(02)02940-3.

volume has a cut plane that separates the two hemispheres. The method described in [23] can be viewed as a dual to that of Mangin *et al.* [17] in that it fills tunnels instead of cutting handles. Both methods, however, can make large, unnecessary changes to the volume, mainly because they only consider one type of correction, either foreground or background, and they have no size criterion.

Deformable surfaces (cf. [24]) have also been used to generate topologically correct cortical surface representations [12], [15], [16]. A parametric deformable surface model guarantees that the topology of the final surface is identical to that of the initial one. (We note that since a parametric surface can develop self-intersections, the topological correctness of the corresponding volume implied by the surface is still not guaranteed.) The problem then becomes one of how to generate a topologically correct initial surface that is close enough to the cortex so that the deformable surface will correctly converge to the cortex. Xu *et al.* [15] used an adaptive fuzzy c-means algorithm to get an initial segmentation of the white matter, which was then successively filtered with median filters until its isosurface had the correct topology. There are two problems with this approach. First, it is not guaranteed to converge. Second, it changes the entire white matter volume when there may only be a local problem. We have found that this method can generate new handles, sometimes very large ones, while working to correct smaller handles in other regions of the volume.

Geometric deformable surface models have also been used to generate cortical surfaces [25]. While there are many advantages in the use of geometric deformable models over parametric ones, their topological flexibility is a liability rather than an advantage in the brain mapping application. The method in [25], for example, finds the inner and outer gray matter surfaces simultaneously, but cannot guarantee their topological correctness. Recent development of topology preserving geometric models [26], [27] shows promise for the refinement of a topologically correct initial surface; but does not solve the problem of finding a topologically correct initial surface close enough to the final surface to avoid poor convergence behavior.

When the initial volume segmentation or surface reconstruction has an incorrect topology, a topology correction algorithm is needed to detect and remove handles/tunnels from the initial volume or surface. Two approaches in the literature operate directly on the triangulated surface meshes rather than the underlying digital volumes [11], [28]. The method reported in [11] replaces the manual editing strategy in [14] with an automatic procedure in which handles are detected as overlapping triangles on the surface after it is inflated to a sphere. The handles are then removed by deleting the overlapping triangles. This method is slow because of the surface inflation step, and the removal of overlapping triangles generally removes a much larger part of the surface than is actually necessary in order to achieve the correct topology, i.e., instead of requiring only a small cut to break the handle, a large portion of the handle may be removed. A second surface-based approach described in [28] removes small handles by simulating wavefront propagation within a certain neighborhood surrounding each mesh vertex. A handle that is smaller than the size of the neighborhood is detected when the wavefronts meet. It is then broken by cutting the mesh along a closed path that encloses the handle and re-tessellating the mesh to seal the resulting holes. This correction will always fill the tunnel associated with a given handle rather than cutting the handle itself, which may not be the best strategy, especially with long, thin handles. Furthermore, the method is computationally intensive, and the final correction strongly depends on the vertex used to identify the defect. In short, it is not a desirable overall strategy for the cortical reconstruction application.

Shattuck and Leahy have recently described an algorithm to detect and correct handles in a digital volume [29], [30]. Instead of region-growing or global filtering, their approach examines the connectivity of the binary white matter segmentation to find regions that give rise to incorrect topology. Then, rather than simply removing these regions, their method carefully edits the underlying volume to make the smallest possible changes (within the limits of their overall approach) that will correct the topology. Their method is elegant and effective, but there is still room for significant improvement. First, the authors acknowledge that their "cuts" are not very natural since they can only be oriented along the Cartesian axes. They also describe a particular topological problem in which "slice duplication" is required. Finally, their approach requires 6-connectivity of the digital object, and has not been generalized for any other digital connectivity. This limits the performance and appearance of the final result as well.

In this paper, we develop a new volumetric topology correction algorithm, which we refer to as the *graph-based topology correction algorithm* (GTCA). GTCA removes all handles from a binary object, which in our case is the largest connected component of a white matter segmentation, filled to remove cavities.[1] As in [29], [30], we rely on a connection graph to detect handles, but our graph construction and editing procedures are quite different. Our graph is constructed by using a morphological opening operation, applied to either the foreground or the background, producing both *body* and *residue* parts of the original object. A conditional topological expansion (CTE) procedure, similar to the region-growing procedure of Mangin *et al.* [17], is used to replace residue parts that do not involve handles. A graph is then constructed by analyzing the connectivity of the body and residue pieces, and one or more residue pieces are removed to break the *cycle*s in the graph, thus removing handles in the volume. Since small corrections are desirable, we apply this procedure on both the object and its background starting with a small structuring element to cut "thin" handles and fill "narrow" tunnels. We then increase the scale of the filter by using larger structuring elements, which guarantees that topological defects are always corrected at the smallest possible scale.

Although similar in broad concept to the method of Shattuck and Leahy, our method is quite different in the details. Our method is intrinsically three-dimensional (3-D), and "cuts" are not forced to be oriented along cardinal axes. It does not require the introduction of half-thickness slices, and any (consistent) digital connectivity definition can be used. A final distinction of our approach is that correct topology can be assured through application of either foreground or background filters alone. This procedure would result in either handles being cut or tunnels

[1]Cavities, unlike handles, are background regions completely surrounded by the object; they are easily removed using standard region-growing.

being filled exclusively, which is not a good option in brain mapping, but may be of use in other applications. In the following sections, we give necessary background, describe the algorithm, and provide experimental results that show the overall characteristics and performance of our new method.

We note that a preliminary version of this paper has been published in a conference proceedings [31].

II. Background

In this section, we present some basic notions about 3-D discrete topology that will be used in describing the topology correction algorithm (see [18]–[22] for more details). We also discuss the relation between digital connectivities and isosurface algorithms. This relationship allows us to compute the genus of a digital object by computing the Euler number of its surface representation. Note that in the application of brain cortical surface reconstruction, we desire a topologically correct surface; the topology correction, however, is performed on the volume data. Thus, we must make sure that the isosurface algorithm gives a topologically correct surface representation of the object.

A. Three-Dimensional Discrete Topology

A 3-D digital image $V \subset Z^3$ is defined as a cubic array of lattice points. We follow the conventional definition of *n-neighborhood* and *n-adjacency*, where $n \in \{6, 18, 26\}$. We denote the n-neighborhood of a point x by $N_n(x)$, and the set comprising the neighborhood of x with x removed by $N_n^*(x)$.

An *n-path of length $l > 0$* from p to q in X, where $X \subset V$, means a sequence of distinct points $p = p_0, p_1, \ldots, p_l = q$ in X such that p_i is n-adjacent to p_{i+1}, for $i = 0, 1, \ldots, l-1$. An n-path $p_0, p_1, \ldots, p_l$ is an n-closed path if and only if p_0 is n-adjacent to p_l. Two points $p, q \in X$ are *n-connected* if and only if there exists an n-path from p to q in X. The set X is called *n-connected* if every two points $p, q \in X$ are n-connected in X. An *n-connected component* of X is a nonempty n-connected subset of X that is not n-adjacent to any other point in X. We denote the set of all n-connected components of X by $\mathcal{C}_n(X)$.

In order to avoid a connectivity paradox, different connectivities, n and $\bar{n}$, must be used in a binary image comprising an object (foreground) X and a background $\bar{X}$. For example, (18, 6) and (26, 6) are two pairs of compatible connectivities. The following definitions are from [21]

Definition 1 (Geodesic Neighborhood): Let $X \subset V$ and $x \in V$. The *geodesic neighborhood* of x with respect to X of order k is the set $N_n^k(x, X)$ defined recursively by: $N_n^1(x, X) = N_n^*(x) \cap X$ and $N_n^k(x, X) = \cup\{N_n(y) \cap N_{26}^*(x) \cap X,\ y \in N_n^{k-1}(x, X)\}$.

Definition 2 (Topological Numbers): Let $X \subset V$ and $x \in V$. The *topological numbers* of the point x relative to the set X are: $T_6(x, X) = \#\mathcal{C}_6(N_6^2(x, X)), T_{6^+}(x, X) = \#\mathcal{C}_6(N_6^3(x, X)), T_{18}(x, X) = \#\mathcal{C}_{18}(N_{18}^2(x, X))$, and $T_{26}(x, X) = \#\mathcal{C}_{26}(N_{26}^1(x, X))$, where # denotes cardinality of a set.

Topological numbers are used to classify the topology type of a lattice point; their efficient computation is described in [21].

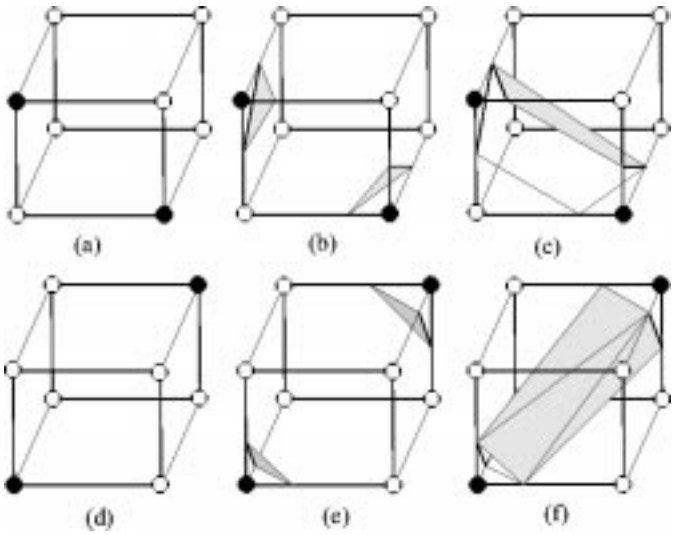

Fig. 2. (a) An ambiguous face; (b) and (c) two possible tilings. (d) An ambiguous cube; (e) and (f) are two possible tilings.

We note that in the definition of topological numbers there are two notations for 6-connectivity. This follows the convention introduced in [21], wherein the notation "6+" implies 6-connectivity whose dual connectivity is 18, while the notation "6" implies 6-connectivity whose dual connectivity is 26. This distinction is needed in order to correctly compute topological numbers under 6-connectivity, and does not imply a different definition of connectivity.

Finally, we introduce the notion of handles. An object X has a *handle* whenever there is a closed path in X that cannot be deformed through connected deformations in X to a single point. For example, a solid torus has exactly one handle. In digital images, the existence of handles depends on the chosen connectivity n. The same explanation of handles is applicable to surfaces, which are two-dimensional objects in 3-D space. A digital object X and its complement $\bar{X}$ have exactly the same number of handles assuming that a pair of compatible connectivities is used. Therefore, we can always say that the tunnel associated with a handle of X is a handle in $\bar{X}$. The number of handles of a digital object is called the *genus* of the object.

B. Marching Cubes (MC) Isosurface Algorithm

The MC algorithm is a standard isosurface algorithm that produces a triangulated surface whose vertices lie on the edges of the cubic lattice [32]–[36]. The MC algorithm processes one cube at a time, and given an isovalue or threshold, it first labels the cube vertices as either inside (higher than the isovalue) or outside. If the surface is determined to pass through the cube, then the MC algorithm estimates the location of the isosurface intersections with the edges of the cube through linear interpolation. These intersections are then connected to form triangles comprising the isosurface itself.

The way in which an isosurface intersects a cube is not always unique. One or more faces or the entire body of the cube can be ambiguous, as shown in Fig. 2. Most MC algorithms used fixed rules—such as always separating "outside" points—to tile these cubes, but this can result in topological paradoxes. As described

in [33] and [36], so-called *face saddle points* and *body saddle points* can be used to produce isosurfaces that are consistent with trilinear interpolation, eliminating these paradoxes.

In this paper, we use a saddle point MC algorithm to produce a surface representation of a binary object ($1 =$ object, $0 =$ background). Note that selection of the isovalue, which can take on any real value in the range $[0, 1]$, due to the interpolation implicit in the isosurface algorithm, has a direct effect on the underlying digital connectivity that the algorithm can represent. For instance, it is straightforward to verify that standard MC algorithms, which do not consider saddle points, can only be made consistent with 6-connectivity by choosing an isovalue greater than 0.5. In order to be consistent with other connectivities, it is necessary to consider both face and body saddle points. Furthermore, following the computation of face/body saddle points as in [36], it is easy to see that for 0–1 valued binary volumes, to yield 26-connectivity we must choose an isovalue less than 0.25; to yield 18-connectivity we must set the isovalue between 0.25 and 0.5; and to be consistent with 6-connectivity, we need an isovalue above 0.5. All of these object connectivities have consistent background connectivities according to the previous discussion.

C. Topology of Surface Meshes

The genus g of a surface mesh is computed from the *Euler number* χ as follows [37]

$$g = 1 - \chi/2 \tag{1}$$

where $\chi = V - E + F$ and V, E, and F are the number of vertices, edges, and faces, respectively, of the surface mesh. In this paper, due to preprocessing steps we need only consider surface meshes consisting of a single connected set of vertices. Such a surface is topologically equivalent to a sphere when $g = 0$. However, neither the Euler number nor the genus provides information about the size or location of a handle.

Given a topologically consistent isosurface algorithm, there is a one-to-one correspondence between the handles on a binary digital object with n-connectivity and that of its triangulated surface representation. Thus, if we remove all the handles on a binary digital object, its boundary surface will be homeomorphic to a sphere. Since direct computation of the genus of a digital object is difficult, and since our final objective is a surface representation of the brain cortex, we always check the topology of the object by computing the Euler number of its surface extracted using the correct MC algorithm. To save time, our MC algorithm can omit the computation of triangle positions when only the Euler number is needed.

III. Graph-Based Topology Correction Algorithm

Our GTCA aims to remove all the handles in a binary volume. We assume that the binary volume has a single connected foreground object with no cavities. A block diagram of the algorithm is shown in Fig. 3(a). There are two types of filters that

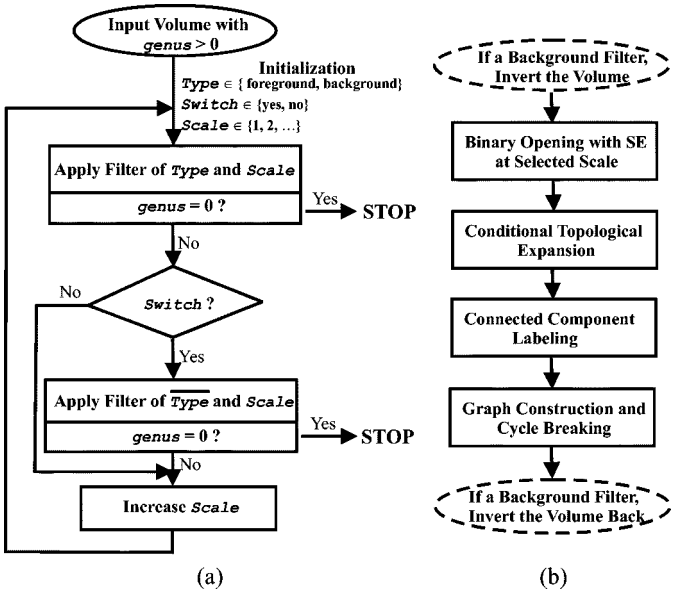

Fig. 3. Topology correction algorithm. (a) Flowchart of the algorithm. (b) Flowchart of a foreground/background filter.

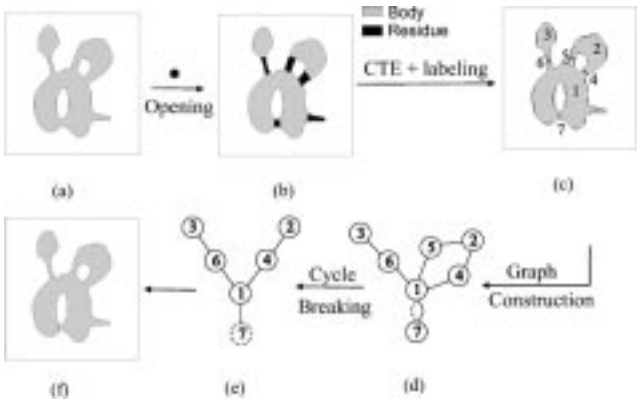

Fig. 4. Illustration of the basic idea.

GTCA can use to correct the topology of an input volume: foreground filters and background filters. Handles removed by a background filter correspond to tunnels filled in the original volume. As shown in Fig. 3(a), GTCA sequentially applies one or both types of filters at successively increasing scales until all the handles or tunnels in the original volume are removed. We note that compatible connectivities must be used for the two filters, which yields an n-connectivity foreground filter and an $\bar{n}$-connectivity background filter. Fig. 3(b) shows the flowchart of a foreground or background filter. The background filter is the same as the foreground filter except that it works on the complement of the original volume and assumes an $\bar{n}$-connectivity instead of the n-connectivity used in the foreground filter. As shown in Fig. 3(b), each filter consists of four major steps. We now describe each step in detail. We refer to Fig. 4 as an illustration of the basic idea that is behind the development of each step. We note that certain parts of the algorithms presented in this section could, in principle, be implemented in parallel. However, we have not considered the implications of parallel implementation and, therefore, the algorithms should be viewed as sequential.

A. Binary Morphological Opening

We use morphological opening as a multiscale analyzer to detect handles at different scales. The morphological opening of object F with structuring element B removes all parts of F that are smaller than B, in the sense that they cannot contain any translated replica of B. We call the structuring element (SE) used at the smallest scale (scale 1) the *basic structuring element*. The structuring element at the scale k is obtained by $k-1$ successive dilations of the basic structuring element with itself. In practice, we use a digital ball of radius one—i.e., an 18-connected neighborhood plus the center point—as the basic structuring element. This gives the ability to make very small topological corrections that are more isotropic than the slightly smaller 3-D cross (another logical choice for the basic structuring element). The shape of the basic SE is not critical to the success of the algorithm although the computation time and the final result will be different. For example, a 3-D cross, which has only seven points, might yield a result with fewer changes to the digital volume, but it might also take longer to compute because it would require more scales. In any case, the resulting filtered volume will have the correct topology.

As illustrated in Fig. 4(b), the opening operation divides the foreground object into two classes. Points that are in the resulting (opened) image are called *body* points, and points removed by the opening operator are called *residue* points. If there are regions within a handle that cannot fit the structuring element, then the handle is broken into body and residue parts. Other parts of the object, such as fingers (see Fig. 1), can also be removed or broken into different body and residue parts as well. The next step of our method seeks to transfer as many points as possible from the residue back to the body without adding a handle.

B. Conditional Topological Expansion

Examination of Fig. 4(c)–(f) leads to the idea that handles might be broken by detecting cycles in a graph comprising body and residue pieces and discarding the smallest residue piece in each cycle. Unfortunately, on a complicated shape such as a white matter segmentation, morphological opening removes far more voxels than just those required to break the handles, no matter what structuring element is used. Typically, the residue set comprises many connected components, several of which are large, complicated shapes. Also, the opening can actually *create* handles in the body component. Thus, introducing a graph-based cycle-removal procedure at this stage is likely to discard unnecessarily large pieces of the object and generate new topological problems.

Our solution is to transfer as many points as possible from the residue back to the body, without introducing handles. Specifically, we grow the body set by successive dilations, but only add those points, which we refer to as *nice points*, that come from the residue and do not adversely affect the topology. A point is *nice* if when added to the body it neither replaces nor creates a handle on the body. We refer to the entire procedure as *conditional topological expansion*. Before describing the algorithm, we first explain the criterion for defining *nice* points.

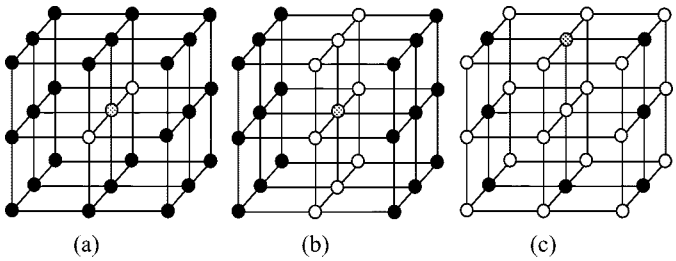

Fig. 5. Illustration of the nice point criterion. In each panel, the gray point represents the residue point under consideration, the dark points represent body points, and the white points represent background points. The gray point is nice in (a) and not nice in (b) and (c).

Whether a point is nice or not can be decided locally using a criterion similar to that of *simple points* in the digital topology literature [18]–[22]. Whereas the addition of a simple point to the body is guaranteed to preserve the exact topology of the body, the addition of a nice point is allowed to fill a tunnel in the body. The use of the nice point concept is necessary because morphological opening can introduce tunnels in the body, and we want CTE to be able to fill them back up. While a simple point must preserve two topological criteria, a nice point only need preserve one criterion. Denote the set of body points by X, and let $T_n(x, X)$ be the topological number as defined in Section II. From [21, p. 1010, Property 5], we have the following property:

Proposition 1 (Nice Points): Let $x \in \bar{X}$ and suppose we add x to X. If $T_n(x, X) = 1$, then it is equivalent to say that the n-components of X are preserved and no n-handles/tunnels are created in X (and no $\bar{n}$-handles/tunnels are created in $\bar{X}$).

Therefore, if x is a residue point that is n-adjacent to the body X, then it is *nice* if and only if $T_n(x, X) = 1$. Fig. 5 illustrates some examples of both nice points and nonnice points. Fig. 5(a) shows a case where a spurious tunnel has been created by a morphological opening. Since the gray (residue) point is originally part of the object, it should be added back to close the "false" tunnel. Note that because the gray point will change the topology when added to the body, it is clearly not a *simple* point.

By adding all the nice points back to the body, we can fill in all the false tunnels created by the opening operation. The cuts on true handles also become thinner; but they never disappear since the condition $T_n(x, X) = 1$ guarantees that adding x does not create handles in the new set $x \cup X$. CTE is performed using the following iterative procedure

Algorithm 1 CTE:

1) Find the set S of residue points that are n-adjacent to the body point set X.
2) For each point $x \in S$, if $T_n(x, X) = 1$, then label it as a nice point.
3) Find and label all the connected components in the set of nice points.
4) Take the largest connected component, and relabel the points in it as new body points.
5) If no point changed its label in Step 4, then stop; otherwise, go to Step 1.

Notice that in Steps 3 and 4 we only grow the largest connected component of nice points at each iteration instead of

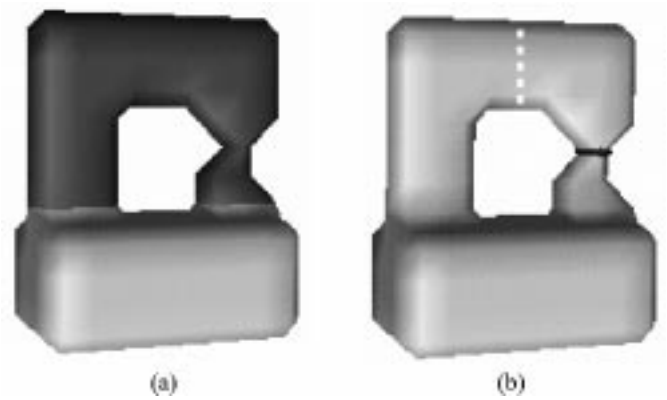

Fig. 6. Illustration of CTE on a simple handle (surface rendering). (a) Original body (light) and residue (dark) after opening. (b) Final cut (dark) after CTE. The dashed line illustrates a nonoptimal cut position.

adding all of the nice points found in Step 2 to the body. The reason for this is to make sure that the final "cuts"—i.e., the final residue points that are not added back to the body—are positioned at the thinnest parts of the handles. This is illustrated in Fig. 6. In Fig. 6(a), the darker shaded piece represents the residue after a morphological opening operation while the lighter shaded piece represents the body. The dotted line in Fig. 6(b) illustrates where the final cut would be if all the nice points were added back to the body after each conditional dilation. The dark line illustrates where the actual cut is placed using CTE. We note that CTE can be made computationally efficient by using a first-in-first-out queue data structure [38].

C. Connected Component Labeling

After CTE, any tunnels in the body that were created by morphological opening are filled in, and the remaining residue pieces form thin "cuts" that separate body components. This situation is illustrated in Fig. 4(b)–(c). Removing all the remaining cuts would certainly eliminate all the handles at this scale, but would also disconnect large portions of the body that we refer to as *fingers*, such as the parts labeled "6" and "3" in Fig. 4(c). Detecting and removing cuts that belong to handles and rejoining to the body those that are not part of handles is the aim of the graph analysis to be described in Section III-D. It turns out, however, that without further analysis of the remaining residue pieces, graph analysis might lead to unnecessary removal or, worse, the accidental inclusion of handles. In this section, we prepare for graph analysis by grouping the body and residue points into sets of connected components with distinct labels. We then compute the number of connections between each pair of body and residue components. When multiple connections are found, they are broken in order to break the corresponding handle(s). Special care is taken both to detect and break handles inside a residue component and to merge suitable residue components in order to avoid false cycles in the subsequent graph analysis. The overall processing strategy described in this section is illustrated in Fig. 7. We note that the final set of body and residue components become the nodes for the graph analysis described inSection III-D.

We begin by labeling the connected components of body points using n-connectivity; these form the *body-connected components* (BCCs). Each residue point that forms a handle by itself with a single BCC is then removed (changed to background). These are the residue points x for which $T_n(x, X) > 1$ for some BCC X [see Fig. 5(c) for an example]. This prevents residue-connected components (RCCs) that are merged to the body later from forming handles that could not be detected by graph analysis. In some unusual configurations, a point removed in this step would not have actually formed a handle in the merged body. In this case, however, the last step of our algorithm (another CTE after graph analysis) adds these points back to the body.

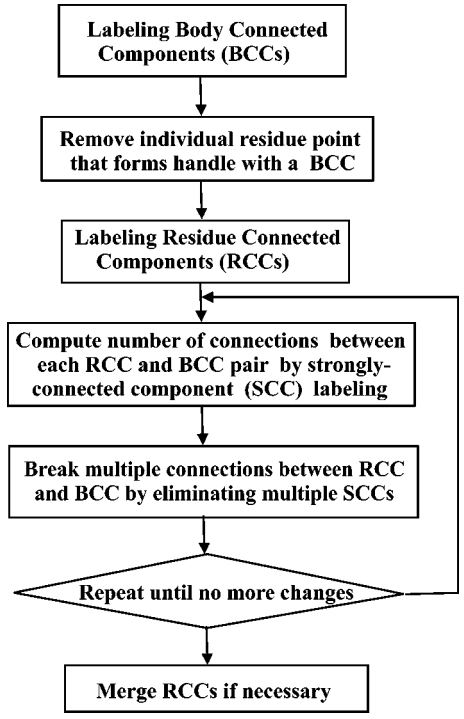

Fig. 7. Connected component labeling and connection analysis.

Next, we compute the n-connected components in the remaining residue points; these form the *residue-connected components*. We must now compute the number of connections between each RCC and each of its adjacent BCCs. This computation is important because an RCC being connected to a BCC more than once indicates that there is at least one handle formed between them [e.g., see the nodes labeled "1" and "7" in Fig. 4(d)]. The way we define the number of connections between an RCC and an adjacent BCC can be considered as a generalization of the definition of topological numbers, which characterize the number of connections between a single point and its adjacent object components. This generalization leads to the concept of *strong connectivity* and *strongly connected component (SCC) labeling* as will be presented later. A subtlety exists when a handle is entirely contained within the RCC itself, as illustrated in Fig. 8. In this case, calculation of the number of connections between the RCC and its adjacent BCCs does not reveal the "hidden handle". The definition of a *weak connectivity* is aimed to resolve this problem. We found that if a handle is contained fully in the RCC but not in the BCC as in Fig. 8 (otherwise, the handle is not yet broken by the opening at the current scale), then there will be two neighboring points in the

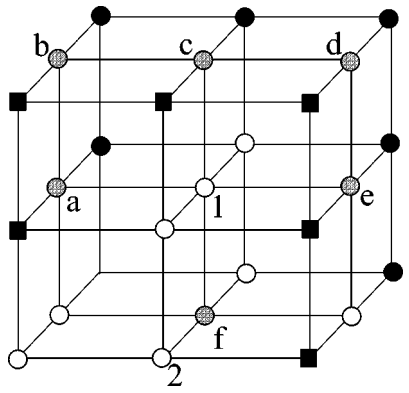

Fig. 8. Illustration of the strong and the weak connectivity definitions ($n = 18$). The gray points belonging to one RCC, the dark disks and the squares represent points belonging to two separate BCCs, and the rest are background points. It can be seen that a handle exists in the RCC.

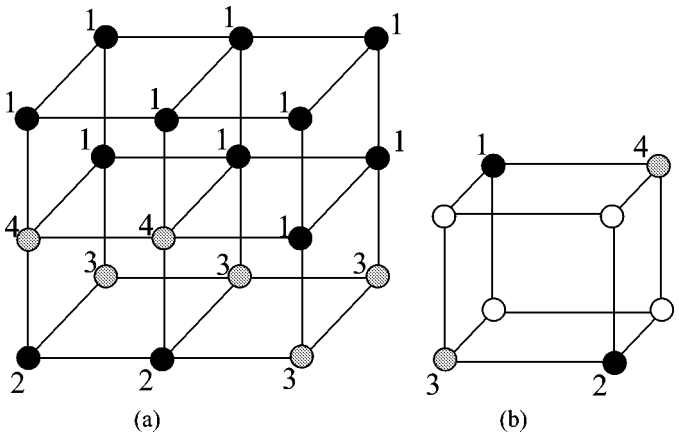

Fig. 9. Necessity of merging RCCs. (a) 6-connectivity and (b) 18-connectivity.

RCC that are not *weakly connected* but would be assigned to the same SCC (e.g., points with label a and f in Fig. 8). Thus, by removing one point from each nonweakly connected pair of neighboring points, we break handles within an RCC. We first present the two definitions (which are new as far as we know):

Definition 3 (Strong Connectivity): Let $X, Y \subset V$. Two points $x_1, x_2 \in X$ that are n-adjacent to Y are `strongly` n_k-`connected` with respect to Y if $x_1 \in N_n^k(x_2, X)$ (or equivalently, $x_2 \in N_n^k(x_1, X)$) and their geodesic neighborhoods of order k inside Y intersect each other, i.e., $N_n^k(x_1, Y)$ and $N_n^k(x_2, Y)$ share at least one common point.

Definition 4 (Weak Connectivity): Let $X, Y \subset V$. Two points $x_1, x_2 \in X$ that are n-adjacent to Y are `weakly` n_k-`connected` with respect to Y if there exists another point x_3 in X that is strongly n_k-connected to both x_1 and x_2.

In the above definitions, k is chosen with respect to n as in the definition of topological numbers. For example, if n is 6^+, then k is 3. We note that strong connectivity implies weak connectivity.

An illustration of the above definitions is shown in Fig. 8. Suppose we consider the strong or weak connectivity relationship among the gray RCC points with respect to the BCC consisting of all the dark squares, and assume that the digital connectivity rule is $n = 18$. It is easy to check that the pairs such as $(a, b), (b, c), (c, d), (d, e)$, and (e, f) are all strongly connected pairs of residue points. Points a and f are 18-neighbors to each other and both 18-adjacent to the BCC, but they are not weakly connected by the above definition. This signals a hidden handle inside the RCC itself. We can break the handle by removing either point a or f to background. Note that if the point with label 1 is also a residue point, then the handle does not exist, but then a and f become weakly connected. Another variation is when point 2 is a foreground point and, thus, belongs to the BCC of dark squares. In this case, the handle is contained in the BCC also, which says that the handle is not yet broken at the current scale. Whether we can detect the handle in the RCC is no longer a relevant issue.

Now let R_i be an RCC and B_j be a BCC that is n-adjacent to R_i. The number of connections between R_i and B_j is defined to be the number of *strongly connected components* (SCCs) formed with respect to B_j by the points in R_i that are n-adjacent to B_j. There are handles formed between R_i and B_j if R_i is connected to B_j more than once or, equivalently, if the points of R_i that are n-adjacent to B_j form more than one SCC under the strong connectivity definition. Thus, after SCC labeling, if multiple SCCs exist within the RCC, we remove all but the largest one in order to break the handles formed between the RCC and the BCC under consideration. As noted above, during the SCC labeling process, we use the weak-connectivity criterion to detect hidden handles, and break them when found.

In particular, we perform the *strongly connected component labeling* (in an RCC R_i with respect to a BCC B_j) in the following way (note that we need only consider the points of R_i that are n-adjacent to B_j). We start with an unlabeled point in R_i that is adjacent to B_j and assign to it a new SCC label. Consider all its n-connected neighbors that are adjacent to B_j. Add a neighbor to the SCC if it is strongly connected. If the neighbor is not weakly connected then relabel it as background. Repeat with another neighbor of the SCC until no more neighbors can be added into it. Then, start from the beginning again to grow another SCC if possible. We remark that there is some arbitrariness in the specific result of the above algorithm as to which point to remove in a nonweakly connected pair of neighbors. Thus, the result of this algorithm is not guaranteed to be unique.

If more than one SCCs exist in R_i with respect to B_j—i.e., R_i is connected to B_j more than once—we keep only the largest SCC, removing all the others by turning them into background. After removing all but the largest SCC, each RCC is now connected to a BCC only one time at most. This step breaks the simple cycles formed by one BCC and one RCC in a connection graph like the one between BCC1 and RCC7 in Fig. 4(d). Since modifying an RCC with respect to one BCC may change the connection between the RCC with other BCCs, we repeat the SCC removal until no more changes are made. This procedure can also split an RCC into several disjoint parts. If this happens, we assign additional labels to each of the new residue components.

Before proceeding to graph-based handle detection, it is necessary to consider the merging of two or more RCCs. The reason for this is illustrated in Fig. 9(a), where $n = 6$ is assumed. In this figure, the dark points belong to the body while the gray points belong to the residue. The numbers beside the points indicate

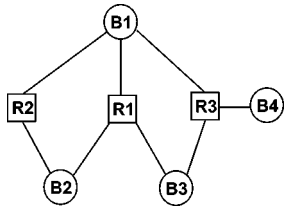

Fig. 10. An example of a connection graph. Suppose RCC R1 is larger than R2 and R3.

the respective BCC/RCC labels. If a graph were formed having one node per label and one edge per connection, the graph would contain the cycle 1-3-2-4-1. Graph analysis would require that either RCC 3 or 4 be removed in order to break the cycle, an apparent handle. Yet, careful scrutiny of the figure shows that in reality the body and residue points taken together form a solid object without any handles. No residue need be removed since there is no handle to break. A similar situation can occur in the case of 18-connectivity, as illustrated in Fig. 9(b).

To resolve the problem, we merge the RCCs that represent the same cut between two body components. Two RCCs are said to represent the *same cut* with respect to two BCCs if the two RCCs are 18-connected at the face (if $n = 6$) or 26-connected at the cube (if $n = 18$) where the separation between the two BCCs occurs, such as RCC3 and RCC4 in Fig. 9(a) or 9(b). It is easily seen that in such cases, there can be no background path passing through the junction where the four components (two RCCs and two BCCs) meet. Thus, the two RCCs need to be merged as one single RCC since the cycle otherwise formed by the four components does not represent a true handle. In our implementation, we merge two RCCs R_i and R_j if there are two points $x \in R_i$ and $y \in R_j$ such that $x \in N_{26}(y)$ and x and y have two common n-neighbors that belong to two distinct BCCs. An RCC can be merged with more than one RCCs if it satisfies this criteria with each one of them.

D. Graph Construction and Cycle Breaking

After considerable preparation, we are now in position to build a graph whose nodes represent the RCCs and BCCs and whose edges represent the connections between them, as illustrated in Fig. 4(d). Note that at this stage, there can be at most one edge between an RCC node and a BCC node. That is, cycles (handles) involving one RCC and one BCC with two or more edges between them, like BCC1 and RCC7 in Fig. 4(d), are already broken in the previous step by removing multiple SCCs. The remaining handles appear as cycles in the graph that are formed by more than two nodes. The graph analysis in this step aims to find such cycles and break them by removing one node from each cycle. Because of the CTE, it is reasonable to assume that the RCCs are much smaller in size than the BCCs. Accordingly, our strategy is to remove one RCC from each cycle in order to break it, which breaks the corresponding object handle in the volume. Since we break cycles by removing *nodes* from the graph rather than *edges* as in [29], [30], the maximum spanning tree algorithm (see [38]) cannot be applied in our approach. A different strategy must be used.

One strategy, which we employed in our earlier work [31], is to identify a cycle (in random order) and break it by removing the smallest RCC belonging to it. This simple approach, however, can result in unexpectedly large changes to the volume. For example, in the case illustrated in Fig. 10, R2 would be removed first if the cycle R1-B1-R2-B2-R1 were found first and R2 were smaller than R1. Later, R3 must be removed in order to break the cycle R3-B1-R1-B3-R3. Unfortunately, these steps cause B4 to be disconnected from the other BCCs, and this might correspond to a very large change to the volume if B4 is large. It is clear from the figure that we could have actually removed R1 to break both cycles, and no BCC would be disconnected.

A better strategy for breaking cycles is based on building cycle-free subgraphs. In this approach, we start by identifying RCCs that we cannot remove without disconnecting a BCC. R3 in Fig. 10 is such an RCC, for example, since its removal would disconnect B4. We refer to these RCCs as *leafnode*s of the graph. The key observation here is that B4 is connected to the rest only through R3. Thus, instead of removing RCCs from the original graph to break cycles, we try to build a maximum subgraph without any cycles (a maximum subtree) starting from leafnodes.

In the following algorithm, the RCCs and the BCCs are labeled separately. The label of an RCC is taken from the set $\{\mathsf{unvisited}, \mathsf{visited}, \mathsf{deleted}, \mathsf{leafnode}\}$. The label of a BCC is a number that indicates a subtree label.

Algorithm 2 (Subtree Growing Algorithm):

0. Label all the RCC nodes of the graph as `unvisited`, set $\mathsf{subtreelabel} = 0$.
1. Find each RCC node that is the only `unvisited` RCC node connected to a BCC node, and relabel it as a `leafnode`.
2. If all the RCC nodes are labeled as either `visited` or `deleted`, then stop. Otherwise, find the largest `leafnode` R_i. If there are no `leafnode`s, find the largest `unvisited` RCC node R_i. Relabel R_i as `visited`.
3. Check the subtree labels of all the BCCs $B_j, j = 1, 2, \ldots, m$ that are connected to R_i. If none of them are assigned a subtree label yet, then increment `subtreelabel`, and assign it to $B_j, j = 1, 2, \ldots, m$. If only one of them is labeled, then assign its subtree label to the other B_js. The remaining possibility is that two or more of the B_js are labeled and have distinct subtree labels. In this case, merge these subtrees as one, and assign (or reassign) the merged label (e.g., the smallest label) to all the B_js.
4. For each `unvisited` or `leafnode` RCC, check whether it has two or more connected BCCs that have the same subtree label. If so, the RCC forms a handle with this subtree and is relabeled as `deleted`.
5. Go to Step 1.

After subtree growing, the new object is reconstructed by putting together all the BCCs and RCCs except those RCCs that are labeled as `deleted`. It is easy to check that this algorithm correctly removes R1 in the graph in Fig. 10, and our experience is that it correctly handles the vast majority of cases. In rare occasions, however, it can happen that more than one subtrees exist after the algorithm stops. In this case, the original

object is split into disjoint parts with one or more `deleted` RCCs connecting them. Ideally, one would like to connect the subtrees so that no body component is lost; but the development of a systematic approach for this process turns out to be a challenging task. It involves splitting one or more `deleted` RCC into appropriate pieces, some of which are removed to break the handles while others are kept so that the subtrees can be jointed together. We have no algorithm for doing this at present. Instead, for these cases we simply remove all the `deleted` RCCs and retain the largest resulting subtree as the reconstructed object. In our experiments, this situation occurs very infrequently and this simple strategy has never brought about a drastic change to the corrected volume. We also note that removing a whole RCC node can sometimes remove good points if the points do not belong to the edges of the cycle to be broken. We run a final pass of CTE after the graph analysis to bring these points back, which will be discussed in Section III-E.

E. Final Stages

There are several steps that must be performed after graph reconstruction. First, it is usually possible to append points that were unnecessarily removed during the connected component labeling and the graph analysis. To do this we apply another pass of CTE using Algorithm 1 with the new object as the body set and the removed points as the residue set. Second, if the present filter is a background filter, the volume is inverted so that background becomes foreground and vice versa. Third, we apply the saddle point MC algorithm to this volume using an isovalue corresponding to the selected connectivity. Last, we compute the genus of the resulting surface. This finishes one pass of the foreground or background filter. If the genus is zero, then the topology of the new volume is correct, and we stop. Otherwise, the proposed algorithm either switches to the opposite filter (background vs. foreground) at the same scale (if not already applied) or increases the scale of the current filter, and repeat the topology correction on the new volume [see Fig. 3(a)]. The algorithm is guaranteed to converge because at some scale, the morphological opening must completely break all the handles, and the CTE and the graph algorithms that follows can then produce a handle-free new object.

IV. Results

A. Simple Demonstration

We applied our algorithm to the piece of white matter shown in Fig. 11(a). Application of a foreground filter with $n = 18$ yields the object shown in Fig. 11(b), while a background filter with $\bar{n} = 6$ yields the object in Fig. 11(c). The foreground filter removed the handle by breaking it along a thin part, while the background filter filled the tunnel with a thin sheet. In both cases, the "cuts" are small and clearly not oriented in Cartesian directions. Fig 11(d)–(h) depicts the foreground cut (black) on a succession of slices through the object.

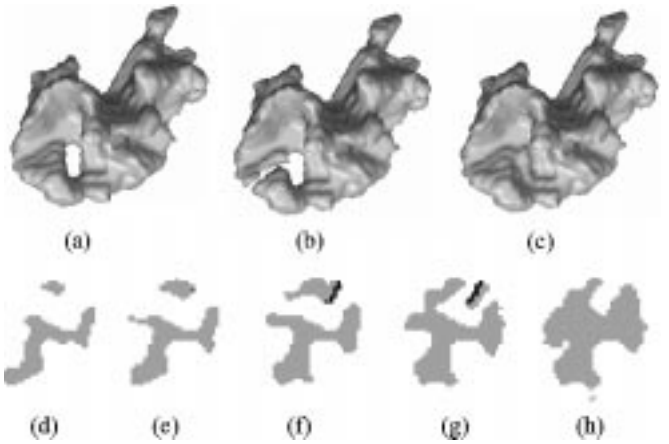

Fig. 11. (a) A handle taken from an actual white matter volume. The result of using (b) a foreground filter and (c) a background filter. (d)-(h) Consecutive slices showing the cuts made by the foreground filter.

B. Brain Volumes

We applied GTCA to 15 MR brain image volumes obtained from the Baltimore Longitudinal Study on Aging [39], [40]. The typical image size after cropping the background is $140 \times 200 \times 160$ voxels. All images were preprocessed and segmented using an updated version of the method described in [15]. Several versions of GTCA were applied to these data sets, including alternating foreground and background filters, foreground-only and background-only filters, alternate object connectivity definitions, and both ball-shaped and cross-shaped basic structuring elements. All filters produced the correct topology in the end, but their performances differed (see Tables I–VI).

The results shown in Tables I–VI used 18-connectivity for the foreground, 6-connectivity for the background, and the 18-connected digital ball for the basic SE. The first row of the tables (under the headings) shows the original genus (number of handles) of the binarized white matter volumes, and the rows below show the genus after each filter stage until a zero genus is obtained in all image volumes. The left-most entry in each row uses an abbreviated notation to indicate whether the filter is a foreground (f) or background (b) filter and the scale of the structuring element. For example, the notation "f2" means foreground filter with structuring element size 2 (the basic SE dilated by itself once) and "b3" means background filter with structuring element size 3. The second to last row shows the number of voxels changed in the final topologically correct volumes as compared with the original ones. The last row shows the average number of voxels changed per handle (ANCPH), which is the ratio of the total number of voxels changed to the genus of the original volume.

Tables I and II show the results from alternating sequences of foreground and background filters with increasing scale. Comparing the results of the two tables, we see that there are always fewer voxels changed and, thus, better ANCPH when the background filter is applied first. The reason for this is that the background filter assumes 6-connectivity while the foreground assumes 18-connectivity. As a result, narrower "swaths" can be used to fill tunnels in the background. On the other hand, the foreground filter yields a faster *initial* reduction in the genus, which shows that the original white matter segmentations have more thin handles than small tunnels. As shown in the tables, the ratio of the number of voxels changed to the genus of the original volume is around 2.8 for the F-B sequence and less than 2.0 for the B-F sequence. It should be noted that the quality of the

TABLE I
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING A F-B SEQUENCE. $(n, \bar{n}) = (18, 6)$; SE = 18-BALL

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	724	955	1376	744	1031	776	562	886	688	825	986	597	1944	1280	801
f1	4	5	19	0	5	5	1	11	4	0	5	5	16	9	4
b1	0	0	1	-	2	0	0	1	0	-	0	1	0	0	0
f2	-	-	0	-	0	-	-	0	-	-	-	0	-	-	-
Changes	2144	2981	4028	1861	2990	1779	1491	2380	1953	2029	2549	1493	5954	3514	2096
ANCPH	2.96	3.12	2.93	2.50	2.90	2.29	2.65	2.69	2.84	2.46	2.58	2.50	3.06	2.74	2.62

TABLE II
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING A B-F SEQUENCE. $(n, \bar{n}) = (18, 6)$; SE = 18-BALL

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	724	955	1376	744	1031	776	562	886	688	825	986	597	1944	1280	801
b1	46	31	31	39	31	24	16	33	26	23	20	17	57	36	20
f1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
b2	-	-	-	-	1	-	-	0	-	-	-	-	-	-	-
f2	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-
Changes	1371	1915	2526	1434	1984	1352	1049	1576	1257	1493	1717	1051	3812	2477	1498
ANCPH	1.89	2.00	1.84	1.93	1.92	1.74	1.87	1.78	1.83	1.81	1.74	1.76	1.96	1.93	1.87

TABLE III
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING F-FILTERS ONLY. $(n, \bar{n}) = (18, 6)$; SE = 18-BALL

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	724	955	1376	744	1031	776	562	886	688	825	986	597	1944	1280	801
f1	4	5	19	0	5	5	1	11	4	0	5	5	16	9	4
f2	0	0	0	0	0	0	0	0	0	-	1	0	0	0	0
f3	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-
Changes	2218	3058	4755	1861	3050	1893	1504	2705	2083	2029	2879	1538	6421	3830	2234
ANCPH	3.06	3.20	3.46	2.50	2.96	2.44	2.68	3.05	3.03	2.46	2.92	2.58	3.30	2.99	2.79

TABLE IV
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING B-FILTERS ONLY. $(n, \bar{n}) = (18, 6)$; SE = 18-BALL

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	724	955	1376	744	1031	776	562	886	688	825	986	597	1944	1280	801
b1	46	31	31	39	31	24	16	33	26	23	20	17	57	36	20
b2	7	7	3	6	6	1	2	6	4	5	6	3	9	6	5
b3	0	1	0	0	0	0	0	1	0	0	1	0	3	1	1
b4	-	0	-	-	-	-	-	0	-	-	0	-	0	0	0
Changes	1951	2827	3319	1777	2284	1768	1287	2181	1617	1785	1962	1395	4836	2907	2101
ANCPH	2.69	2.96	2.41	2.39	2.22	2.28	2.29	2.46	2.35	2.16	1.99	2.34	2.49	2.27	2.62

TABLE V
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING A B-F SEQUENCE. $(n, \bar{n}) = (18, 6)$; SE = 6-CROSS

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	724	955	1376	744	1031	776	562	886	688	825	986	597	1944	1280	801
b1	130	166	144	98	108	94	66	103	94	108	103	77	249	169	85
f1	2	9	12	4	3	3	7	11	6	3	6	5	8	8	7
b2	1	4	5	3	1	0	3	5	2	0	3	1	3	3	5
f2	0	0	1	0	1	-	0	0	0	-	0	0	0	0	0
b3	-	-	0	-	0	-	-	-	-	-	-	-	-	-	-
Changes	1270	1714	2335	1346	1782	1238	972	1497	1126	1296	1573	993	3734	2136	1283
ANCPH	1.75	1.79	1.70	1.81	1.73	1.60	1.73	1.69	1.64	1.57	1.60	1.66	1.92	1.67	1.60

TABLE VI
GENUS, NUMBER OF VOXELS CHANGED, AND ANCPH OF USING AN F-B SEQUENCE $(n, \bar{n}) = (26, 6)$; SE = 18-BALL

Brain	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Original	1049	1354	2008	1075	1522	1132	863	1313	979	1201	1438	854	2726	1843	1186
f1	12	11	35	0	13	12	2	14	8	5	12	9	21	11	12
b1	1	0	1	-	2	0	0	2	0	0	0	1	1	0	0
f2	0	-	0	-	0	-	-	0	-	-	-	0	0	-	-
Changes	3046	4004	5559	2612	4080	2541	2115	3240	2642	2817	3516	2051	8221	4655	2859
ANCPH	2.90	2.96	2.77	2.43	2.68	2.24	2.45	2.47	2.70	2.34	2.44	2.40	3.02	2.52	2.41

initial segmentation has a substantial impact on ANCPH. Therefore, although our results are comparable to those in [29] and [30] this does not allow us to conclude that the algorithms are comparable. A direct comparison on the same data is needed.

Tables III and IV show the results arising from the application of only one type of filter, foreground or background, respectively. These tables verify that our algorithm can achieve a topologically correct result by either cutting handles or filling tunnels exclusively. It is also observed that the background filters yield overall fewer changes to the volume than the foreground filters. Once again, this can be attributed to the fact that the background filters assume 6-connectivity rather than 18-connectivity. Comparing these tables with Tables I and II also shows that the topological correction with the smallest modification to the volume is achieved by alternating background and foreground filters. This process guarantees that a topological defect is corrected at the smallest possible scale.

Table V shows the results of using the 3-D cross as the basic SE instead of the 18-ball. Again, the correct topology is achieved for each data set. Comparing this result to that in Table II reveals that the cross has better performance as measured by the total number of voxels changed. This is attributed to the fact that the cross has finer structure than the ball, so defects can be corrected at even finer scales. The tradeoff is that the cross requires five scales instead of four and, therefore, more computation time in order to guarantee topological correctness of all 15 volumes.

Table VI shows the result from application of 26-connectivity to the foreground and 6-connectivity to the background. As expected, GTCA successfully corrects the topology of all 15 brain volumes. Comparison with Table I, however, shows that it does not perform as well as the (18,6) topological pair, as measured by the number of voxels changed in the final volume. It is also interesting to note that use of 26-connectivity yields a much larger genus in the *original* volume than does use of 18-connectivity. These facts taken together suggest that use of (18,6) connectivity rather than (26,6) connectivity will produce a more delicate topological correction.

C. Visualization and Computation

Fig. 12 shows one part of a WM/GM surface before and after topology correction using a background-foreground sequence with 6 connectivity for the background and 18-connectivity for the foreground. The basic structuring element used in this experiment is the 18-ball. All handles seen in Fig. 12(a) are clearly removed in Fig. 12(b).

The processing time of GTCA depends on the total number of foreground/background filters required. For the results shown in Tables I–VI, each filter took less than 3 min on an SGI Onyx2 workstation with a 250-MHz R10000 processor, and the total processing time for each brain volume took less than 10 min.

V. Discussion

We have experimentally shown that our topology correction algorithm works successfully on brain volumes having large numbers of handles. Furthermore, it works successfully with different choices of connectivity rules, different filtering sequences, and different structuring elements. It is clear from the theory that it will always produce a topologically correct final volume; however, we make no claim that our result is optimal by any criteria. Instead, our approach can be considered to be a (presumably) suboptimal solution to the goal of producing a topologically correct object by making the fewest number of changes to the voxels in the original volume.

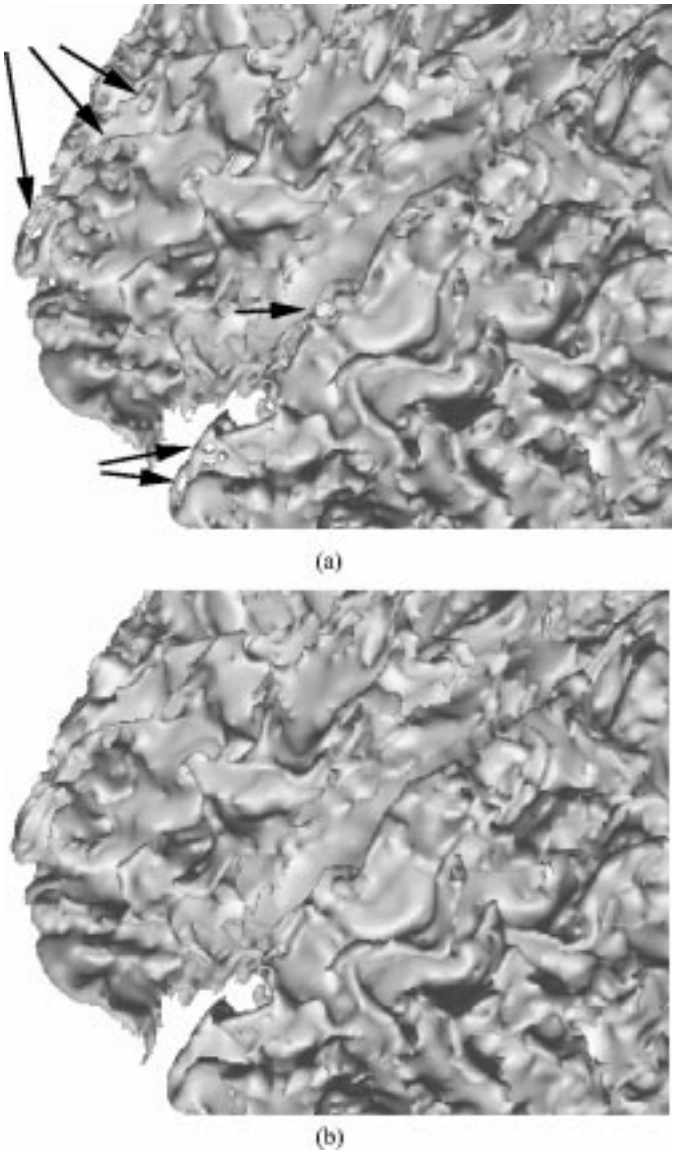

Fig. 12. Cortical surfaces obtained (a) before and (b) after topology correction.

It might be asked, however, whether the criterion of making the fewest number of changes to the original volume is a sensible one. The primary rationale behind this criterion is that topology defects are caused by image noise, and are primarily of a fine scale. Our strategy to alternate between foreground and background, then go from fine to coarse scales acknowledges the fact that such noise can either "add" or "remove" parts of the object and that noise can sometimes group together to form larger defects.

But this focus and strategy does not necessarily make changes that are concordant with the actual anatomy. For example, suppose two opposing banks of a sulcus were wrongly "bridged" near its gyral lips. The two possible corrections to this topological defect are to remove the bridge or to fill the entire sulcus. If the sulcus were very narrow and the bridge were very thick, then it is possible that our algorithm would erroneously fill the

sulcus. Fortunately, this situation is unlikely to exist when correcting the white matter volume, since the sulcal separation includes both the sulcal gap itself and both sulcal banks comprising cortical gray matter. Therefore, it is unlikely that such bridges will exist, assuming a good segmentation algorithm is used. Furthermore, it is much more likely that the scale of any such bridge will be smaller than the scale of the sulcus. In this case, the bridge will be cut first and the correct result will be achieved regardless of the order of background and foreground alternation. On the other hand, if our algorithm were applied to a segmentation comprising both the gray matter and white matter, then an erroneous sulcal filling is more likely. Thus, it is important for the production of anatomically correct results that a good segmentation algorithm be used, and our correction algorithm is applied on the white matter segmentation.

It is conceivable that further improvements to our algorithm can be made by taking anatomical information into account. As a first step, it would be desirable to incorporate either intensity information from the MR images or partial volume information from a fuzzy segmentation into the algorithm. This information might be useful in distinguishing a weak bridge from actual cortical gray matter, for example, preventing the erroneous sulcal filling described above. Also, we might weight the voxels by a suitable function of the image intensity when performing the CTE and the cycle breaking. It may even be possible to develop a fully gray-scale topological correction scheme, rather than relying on a binary segmentation as in our algorithm. More detailed anatomical information could also come from higher-level image analysis steps such as a volumetric registration to a brain atlas which would allow us to label sulci versus gyri, and even to label certain voxels as "anomalous" because they do not register well to labeled parts of the atlas. Such anomalous regions would then be made more susceptible to removal under topological correction. Although we have not identified any specific problems regarding incorrect anatomical corrections using our algorithm, it may be fruitful to explore these ideas for future incorporation should such problems arise after more extensive evaluation.

VI. Conclusion

We have developed and evaluated GTCA, an automatic method to remove handles in 3-D digital images. GTCA is fundamentally based on the theory of digital topology and uses results from mathematical morphology, implicit surface tiling, and graph theory. The advantages of using GTCA are as follows.

1) It is intrinsically 3-D, cuts (respectively, fills) handles (respectively, tunnels) naturally in arbitrary directions.
2) It works with any consistent pair of digital connectivities.
3) It uses morphological opening as a multiscale analyzer so that corrections are made at the smallest scales.
4) A foreground filter or background filter can be used alone to cut handles or fill tunnels exclusively, if desired.

The method has been shown to work very well on 15 MR-segmented volumes. We expect GTCA to be a useful automated tool that can help the processing of large numbers of brain data sets for brain geometric and functional analysis studies across populations.

Acknowledgment

The authors would like to thank Dr. S. Batman, Dr. J. Goutsias, Dr. D. Pham, and Dr. S. Resnick for their contributions. They would also like to thank the anonymous reviewers for their helpful suggestions.

References

[1] R. Kikinis, M. E. Shenton, G. Gerig, H. Kokama, J. Haimson, B. F. O'Donnell, C. G. Wible, R. W. McCarley, and F. A. Jolesz, "Temporal lobe sulco-gyral pattern anomalies in schizophrenia: An *in vivo* MR three-dimensional surface rendering study," *Neurosci. Lett.*, vol. 182, pp. 7–12, 1994.

[2] P. M. Thompson, J. Moussai, S. Zohoori, A. Goldkorn, A. A. Khan, M. S. Mega, G. W. Small, J. L. Cummings, and A. W. Toga, "Cortical variability and asymmetry in normal aging and Alzheimer's disease," *Cereb. Cortex*, vol. 8, no. 6, pp. 492–509, 1998.

[3] M. Vaillant and C. Davatzikos, "Hierarchical matching of cortical features for deformable brain image registration," in *Proc. XVIth Int. Conf. Information Processing in Medical Imaging (IPMI'99)*, pp. 182–195.

[4] X. Tao, X. Han, M. E. Rettmann, J. L. Prince, and C. Davatzikos, "Statistical study on cortical sulci of human brains," in *Proc. XVIIth Int. Conf. Information Processing in Medical Imaging (IPMI'01)*, June 2001, pp. 475–487.

[5] 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.

[6] D. C. van Essen and H. A. Drury, "Structural and functional analyses of human cerebral cortex using a surface-based atlas," *J. Neurosci.*, vol. 17, no. 18, pp. 7079–7102, 1997.

[7] D. C. van Essen and J. H. R. Maunsell, "Two dimensional maps of cerebral cortex," *J. Comparative Neurol.*, vol. 191, pp. 255–281, 1980.

[8] G. J. Carman, H. A. Drury, and D. C. van Essen, "Computational methods for reconstructing and unfolding the cerebral cortex," *Cereb. Cortex*, vol. 5, pp. 506–517, 1995.

[9] B. Wandell, S. Engel, and H. Hel-Or, "Creating images of the flattened cortical sheet," *Investigat. Ophthalmol. Visual Sci.*, vol. 36, p. S612, 1996.

[10] B. Fischl, M. I. Sereno, and A. M. Dale, "Cortical surface-based analysis II: Inflation, flattening, and a surface-based coordinate system," *NeuroImage*, vol. 9, pp. 195–207, 1999.

[11] 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.

[12] C. Davatzikos and N. Bryan, "Using a deformable surface model to obtain a shape representation of the cortex," *IEEE Trans. Med. Imag.*, vol. 15, no. 6, pp. 785–795, 1996.

[13] 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, no. 6, pp. 852–863, 1997.

[14] 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.

[15] C. Xu, D. L. Pham, M. E. Rettmann, D. N. Yu, and J. L. Prince, "Reconstruction of the human cerebral cortex from magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 18, no. 6, pp. 467–480, 1999.

[16] D. MacDonald, N. Kabsni, D. Avis, and A. C. Evans, "Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI," *NeuroImage*, vol. 12, pp. 340–355, 2000.

[17] J.-F. Mangin, V. Frouin, I. Bloch, J. Regis, and J. Lopez-Krahe, "From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations," *J. Math. Imag. Vis.*, vol. 5, pp. 297–318, 1995.

[18] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Comput. Vis., Graphics, Image Processing*, vol. 48, pp. 357–393, 1989.

[19] G. Malandain, G. Bertrand, and N. Ayache, "Topological segmentation of discrete surfaces," *Int. J. Comput. Vis.*, vol. 10, no. 2, pp. 183–197, 1993.

[20] G. Bertrand and G. Malandain, "A new characterization of three dimensional simple points," *Pattern Recogn. Lett.*, vol. 15, pp. 169–175, 1994.

[21] G. Bertrand, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recogn. Lett.*, vol. 15, pp. 1003–1011, 1994.

[22] P. K. Saha and B. B. Chaudhuri, "Detection of 3D simple points for topology preserving transformations with application to thinning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 1028–1032, 1994.
[23] Z. Aktouf, G. Bertrand, and L. Perroton, "A 3D-hole closing algorithm," in *Proc. 6th Int. Workshop Discrete Geometry for Computer Imagery*, France, 1996, pp. 36–47.
[24] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Med. Image Anal.*, vol. 1, no. 2, pp. 91–108, 1996.
[25] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan, "Segmentation and measurement of the cortex from 3D MR images," in *Proc. Med. Image Computing and Computer-Assisted Intervention (MICCAI)*, 1998, pp. 519–530.
[26] X. Han, C. Xu, and J. L. Prince, "A topology preserving deformable model using level sets," in *Proc. IEEE CS Conf. Computer Vision Pattern Recognition (CVPR2001)*, Hawaii, Dec. 2001, pp. II765–II770.
[27] X. Han, C. Xu, D. Tosun, and J. L. Prince, "Cortical surface reconstruction using a topology preserving geometric deformable model," in *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA2001)*, Hawaii, Dec. 2001, pp. 213–220.
[28] I. Guskov and Z. J. Wood, "Topological noise removal," in *Proc. Graphics Interface 2001*, pp. 19–26.
[29] D. W. Shattuck and R. M. Leahy, "Topological refinement of volumetric data," in *Proc. SPIE*, vol. 3661, San Diego, CA, Feb. 1999, pp. 204–213.
[30] ——, "Topologically constrained cortical surfaces from MRI," in *Proc. SPIE*, vol. 3979, San Diego, CA, Feb. 2000, pp. 747–758.
[31] X. Han, C. Xu, U. Braga-Neto, and J. L. Prince, "Graph-based topology correction for brain cortex segmentation," in *Proc. XVIIth Int. Conf. Information Processing in Medical Imaging (IPMI'01)*, June 2001, pp. 395–401.
[32] W. E. Lorensen and H. E. Cline, "Marching cubes: A high-resolution 3D surface construction algorithm," *ACM Comput. Graphics*, vol. 21, no. 4, pp. 163–170, 1987.
[33] G. M. Nielson and B. Hamann, "The asymptotic decider: Resolving the ambiguity in marching cubes," in *Proc. IEEE Visualization 1991*, Los Alamitos, CA, pp. 83–91.
[34] P. Ning and J. Bloomenthal, "An evaluation of implicit surface tilers," *IEEE Comput. Graphics Applicat.*, vol. 13, pp. 33–41, Nov. 1993.
[35] A. van Gelder and J. Wilhelms, "Topological considerations in isosurface generation," *ACM Trans. Graphics*, vol. 13, no. 4, pp. 337–375, 1994.
[36] B. K. Natarajan, "On generating topologically consistent isosurfaces from uniform samples," *Visual Comput.*, vol. 11, no. 1, pp. 52–62, 1994.
[37] M. K. Agoston, *Algebraic Topology—A First Course*. New York, NY: Marcel Dekker, 1976.
[38] R. Sedgewick, *Algorithms in C*. Reading, MA: Addison-Wesley, 1990.
[39] N. W. Shock, R. C. Greulich, R. Andres, D. Arenberg, P. T. Costa, Jr., E. Lakatta, and J. D. Tobin, *Normal Human Aging: The Baltimore Longitudinal Study of Aging*. Washington, DC: U.S. Government Printing Office, 1984.
[40] S. M. Resnick, A. F. Goldszal, C. Davatzikos, S. Golski, M. A. Kraut, E. J. Metter, R. N. Bryan, and A. B. Zonderman, "One-year age changes in MRI brain volumes in older adults," *Cereb. Cortex*, vol. 10, no. 5, pp. 464–472, 2000.
[41] M. Vaillant and C. Davatzikos, "Finding parametric representations of the cortical sulci using an active contour model," *Med. Image Anal.*, vol. 1, no. 4, pp. 295–315, 1997.