Topology Repair of Solid Models Using Skeletons

Qian-Yi Zhou, Tao Ju, Shi-Min Hu

Abstract

We present a method for repairing topological errors on solid models in the form of small surface handles, which often arise from surface reconstruction algorithms. We utilize a skeleton representation that offers a new mechanism for identifying and measuring handles. Our method presents two unique advantages over previous approaches. First, handle removal is guaranteed not to introduce invalid geometry or additional handles. Second, by using an adaptive grid structure, our method is capable of processing huge models efficiently at high resolutions.

Index Terms

topology repair, skeleton, octree

Topology Repair of Solid Models Using Skeletons

I. Introduction

With the advance of data acquisition techniques, we have witnessed a boom of high resolution 3D data in recent years. Although many surface reconstruction methods are capable of generating water-tight surfaces from these data, the resulting models may still exhibit topological errors in the form of small handles, such as those shown on the left of Figure 1. These high-frequency topological features may unnecessarily increase the complexity of the model and make it unsuitable for subsequent processing tasks such as mesh simplification, mesh parameterization, and physical computation.

Our goal is to remove small handles on the surface of a solid model, so that a low-genus model can be prepared for further applications. To be able to process large models with complex errors, which are typical in today's surface reconstruction problems, we particularly desire the following properties:

- **Discriminative:** The method should be able to differentiate between big and small handles.
- **Robust:** Removal of existing handles should not introduce invalid geometry or additional handles.
- **Efficient:** The method should handle huge models at high resolutions within reasonable time and memory.

Unfortunately, to the best of our knowledge, none of the current topology repair methods satisfy all of our requirements. In particular, it is difficult for mesh-based or existing volumetric methods to guarantee that removing a handle does not introduce a new handle. Furthermore, the time and space consumption of traditional methods are typically high for processing large models, either due to operations that require the full mesh resolution or the reliance on a uniform volumetric grid.

In this paper we introduce a new, volumetric approach of topology repair that meets all of the three requirements. Our method thins a 3D volumetric model to a discrete skeleton, where the task of detecting handles on the model is reduced to detecting cycles in a graph defined by the skeleton. Using topology-preserving morphological operations, the modified skeleton with cycles removed grows back into the model with the corresponding handles removed. Both thinning and growing are performed on an adaptive grid structure for efficient processing of large models. In addition, our method allows selective removal of small handles by computing and utilizing a thickness measure on the skeleton.

Contributions We present a robust and efficient solution for topology repair. Our method consists of conceptually simple steps, and possesses two advantages over existing techniques:

1) Unlike previous mesh-based or volumetric approaches, our method is guaranteed to remove handles without introducing additional ones via skeleton modification and topology-preserving morphological operations.
2) Whereas previous volumetric methods rely on a uniform grid structure, our method operates on an adaptive octree grid and is capable of processing huge models at very high resolutions (e.g., 4096^3) in minutes.

II. Related works

A. Skeletons

Skeletons are compact, medial representations that describe the shape and connectivity of a 3D object [1]. In contrast to skeletonization methods based on voronoi diagrams [2], [3] and distance transforms [4], [5], iterative *thinning* excels as an efficient, easy-to-implement technique for generating topology-preserving skeletons of volumetric images (see an excellent survey in 2D by [6] and a 3D introduction by [7]). Given a solid represented as a set of 3D grid points, each thinning iteration removes points in the outmost layer of the set. The key of topology-preservation lies in identifying *simple points* [8] whose removal would not alter the topology of the solid. Unfortunately, existing thinning methods rely heavily on the use of a uniform grid because it is convenient to identify simple points on such a grid, which limits these methods to relatively low resolutions that are often insufficient for capturing the topology of large models.

B. Topology-controlled surface reconstruction

This class of methods [9]–[12] is designed to reconstruct iso-surfaces from volumetric data with a known topology type. Starting with an initial solid with the correct surface topology, these methods grow (as opposed to thin) the solid in a topology-preserving manner. While such methods have been effective in reconstructing topological spherical cortical surfaces from MRI data, application to other topologies is difficult as these methods require *a priori* knowledge of the desired topology as well as the geometry of the initial solid.

C. Mesh-based topology repair

The first class of methods for repairing the topology of a given surface performs surgeries directly on the polygonal mesh. Representative work includes the method of Fischl *et al.* [13], which inflates a reconstructed cortical surface into a sphere and removes handles by identifying and deleting overlapping triangles on the inflated sphere. Using the concept of α-hulls, El-Sana and Varshney [14] achieve controlled simplification of CAD models by identifying small tunnels and surface concavities as regions not accessible to a sphere of user-specified radius rolling on the surface. Also in a controlled

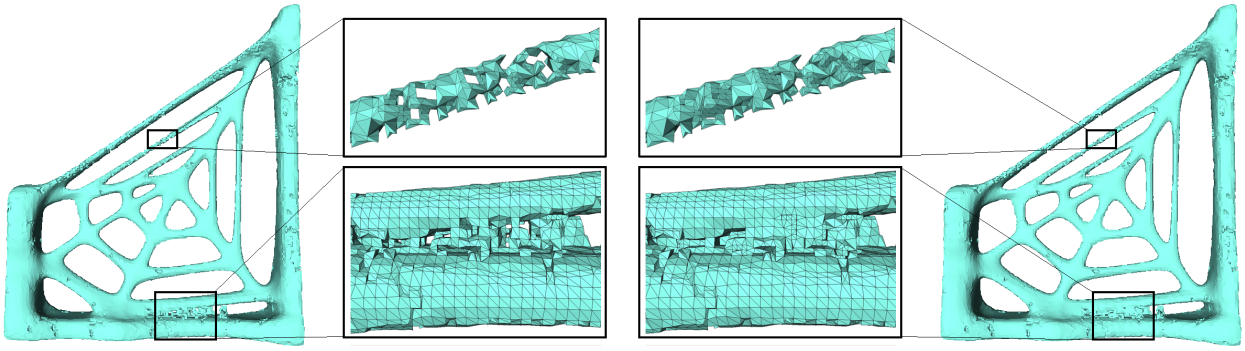


Fig. 1. Topology repair on a spider-web model: the original genus-75 model reconstructed from point sets with many small and entangled handles (left), topology repair removes all erroneous handles except for the 17 major web holes (right).

manner, Guskov and Wood [15] employs a surface growing technique that identifies and removes small handles completely contained in a mesh neighborhood of a given size.

Mesh-based methods have the advantage that topology changes only involve local modification of the geometry. However, there are two typical drawbacks. First, the removal of existing handles directly on the mesh may introduce invalid geometry in the form of self-intersections. Second, it is computationally expensive to identify handles directly on a large mesh, for example, by surface inflation [13], by computing and intersecting α-prisms of triangles [14], or by exploring a surface neighborhood that can be potentially large for identifying long and thin handles [15].

D. Volumetric topology repair

The second class of methods, to which our method belongs, removes surface handles by modifying a volume representation of the input model. In a simple approach, Nooruddin and Turk [16] applied opening and closing operations on the volume to remove small surface handles. However, these global morphological operations may create additional handles in areas away from the existing ones. In a more targeted approach, the method of Wood *et. al.* [17] detects each surface handle as a cycle in the Reeb graph of the iso-surface extracted using the Marching Cubes method [18], and performs handle removal by filling a disk-like volume inside the shortest geodesic loop corresponding to each cycle. Still, the main problem with this hybrid approach, as commented on by the authors, is the possible introduction of new handles due to the loop-filling operation. In addition, the removal of each handle requires re-building of the Reeb graph on a uniform grid, which can be time-consuming for a large number of handles. The detection of shortest geodesic loops on big handles can also be expensive.

Our method is most closely related to the graph-based approach of Shattuck and Leahy [19] and Han *el al.* [20]. Both methods encode the topology of the solid (instead of the surface) as a graph, and remove handles by breaking cycles in the graph. Using topology-preserving morphological operations, handle removals are guaranteed not to introduce new handles. However, both methods involve complex graph generation and analysis that are restricted to uniform grids. In particular, the construction of the Reeb graph in [19] is based on axes-aligned sweeping, while handle removal using the graph in [20] requires non-trivial connectivity analysis to identify "hidden" handles within each graph node. In contrast, our skeleton representation of the solid is simple enough to compute on an adaptive grid and allows for easy identification of surface handles of different sizes.

Recently, a multi-resolution solution was proposed by Szymczak and Vanderhyde [21], which applied topology-preserving carving operations to extract iso-surfaces with the desired genus. This method, however, provides no direct means for controlling the size of the handles to be removed. In addition, the removal operation is limited to filling tunnel-like handles, and hence may result in modifying a much larger volume than necessary. In comparison, our handle removal is guided by an accurate measure of handle sizes, and allows for both tunnel-filling and ring-cutting (see the removal of the two handles in Figure 6 (e)).

III. Method Overview

To avoid introducing invalid geometry (e.g., self-intersections) as the result of topology repair, we represent an input model as an implicit volume. The surface of the model, represented as the iso-surface on the volume, partitions the volume into the *object* (e.g., interior) and the *background* (e.g., exterior). To remove surface handles, our method involves three conceptually simple steps, as illustrated in Figure 2:

1) **Thin** the object into a *skeleton* that preserves the topology of the object (b).

2) **Remove** cycles in the skeleton by computing the spanning tree of the graph defined by the skeleton (c).

3) **Grow** the modified skeleton to form a new object that preserves the topology of the skeleton (d).

Intuitively, a cycle in the skeleton corresponds to a ring-like handle on the original surface, and removing the skeleton cycle has the effect of "cutting" the ring at the location where the

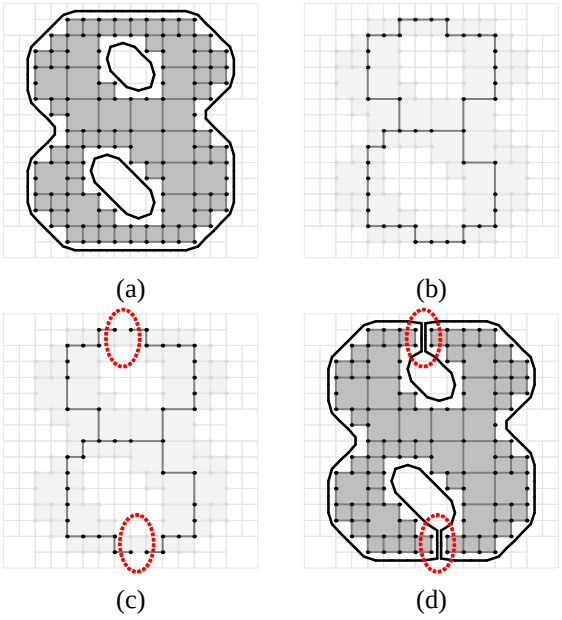


Fig. 2. 2D illustration of handle removal using skeletons. (a): The original object (darkened points, edges and faces) and the iso-surface (solid lines). (b): The skeleton of the object. (c): The modified skeleton consisting of a spanning tree of (b) (removed edges are highlighted). (d): The new object grown from the modified skeleton (c), resulting in the removal of handles.

cycle is cut (see Figure 2 (c,d)). In our method, removing one skeleton cycle is guaranteed to cut exactly one surface handle without introducing additional handles (unlike mesh-based [13]–[15] or previous volumetric [16], [17] handle-removal methods). Furthermore, we can associate the skeleton with a thickness function, which allows the user to control the size of handles to be removed and allows each ring to be cut at its thinnest location.

The above steps can be applied to both the object and the background. When applied to the background, a cycle in the background skeleton corresponds to a tunnel-like handle on the original surface, and removing a skeleton cycle results in "filling" of the tunnel. Like cutting, filling is guaranteed not to introduce additional handles, and tunnels can be selectively filled based on their sizes.

IV. Volume Representation

Before presenting the main algorithms, we first introduce an adaptive volume representation on which the algorithms will be performed.

A. Motivation

Our volume representation is motivated by thinning, the procedure that we will use to compute skeletons (see detailed discussion in the next section). Our thinning algorithm operates on a 3D *cellular complex*, which consists of points (0-D), edges (1-D), faces (2-D) and cells (3-D). In particular, each edge connects two points, each face is enclosed by a ring of edges, and each cell is enclosed by an envelop of faces. For example, the darkened points, edges and faces in each grid of Figure 2 form a cellular complex in 2D.

To represent a cellular complex on a volumetric grid, we need to be able to tag each grid element (e.g., point, edge, face and cell) that belongs to the complex. Note that merely storing signs at grid points, as in traditional volume representations, is not sufficient: the edge connecting two points that belong to a cellular complex may not, itself, be part of that complex (see highlighted region in Figure 2 (c)). Even the more advanced representation [22] restricts tagging to just points and edges.

B. Representation

We begin with an octree structure to support efficient processing of large models at high grid resolutions. We additionally associate each *minimal* grid element of the octree with a $+/-$ sign. Here, a minimal element is the one that does not contain any smaller elements of the same dimension (e.g., a minimal edge contains no smaller edges on the grid). For convenience, we shall drop the prefix "minimal" hereafter. We call the new volume representation an *Extended Signed Octree* (ESO).

To facilitate thinning, both object and background must assume the form of a cellular complex. The *object* V in an ESO G is defined as the set of all positive elements in G. Note that not every ESO yields an object that is a cellular complex: a positive edge containing a negative point violates our previously stated definition that an edge in a cellular complex must contain two points of the complex. As a result, we further require that, in a *valid* ESO grid, each positive element must contain only positive elements of lower dimensions.

Unlike the object, the set of all negative elements on a valid ESO is *not* a cellular complex. To this end, we consider the *dual* of a valid ESO grid G, denoted as $\hat{G}$, which consists of points, edges, faces and cells topologically dual to the cells, faces, edges and points on G. In addition, each element in $\hat{G}$ is given the sign of its dual element in G. Geometrically, points in $\hat{G}$ are located at the centroids of their corresponding cells on the primal grid G.[1] A 2D illustration of a portion of ESO grid and its dual are shown in Figure 3 (a,b). As such, we define the *background* $\overline{V}$ as the set of negative elements in the dual grid $\hat{G}$ (shown as dimmed elements in Figure 3 (b)). Since each negative element of a valid ESO is only shared by negative elements of higher dimensions, by duality, every negative element in $\hat{G}$ contains only negative elements of lower dimensions. Therefore, $\overline{V}$ is also a cellular complex.

Finally, we note that symbols G, $\hat{G}$, V and $\overline{V}$ all refer to a same volume representation. In particular, any changes to the object V involve flipping the signs of some grid elements in the primal grid G, hence affecting the signs in the dual grid $\hat{G}$ and the composition of the background $\overline{V}$. In addition, we note that $\hat{\hat{G}} = G$ and $\overline{\overline{V}} = V$.

[1]For completeness, the outside of the root node of the primal octree is represented as a cell element in G with infinite size, whose dual in $\hat{G}$ is a point at infinity.

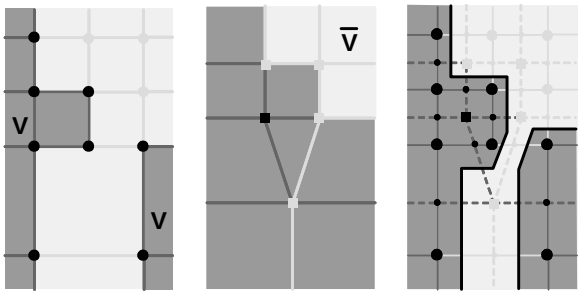


Fig. 3. (a) A primal ESO grid G, (b) The dual grid $\hat{G}$, (c) The composite grid $\tilde{G}$ constructed by overlaying G with $\hat{G}$ and the iso-surface (solid lines). In (a,b), positive grid elements are darkened and negative elements are dimmed.

C. Operations

1) Constructing ESO: A valid ESO grid can be easily converted from a traditional octree grid, where signs are stored at grid points, by retaining existing signs while assigning positive signs to edges, faces and cells that contain only positive points. The initial octree grid can be obtained either directly from a volume image (e.g., MRI data) or from a polygonal mesh using scan-conversion routines. In this paper, we use the PolyMender software [23], which is capable of producing a water-tight solid model from arbitrary polygonal soups.

2) Extracting Iso-surface: To construct the iso-surface of an ESO grid G, we extend the Dual Contouring algorithm [24], which was designed for octrees with only signs at grid points. In particular, we consider a composite grid, denoted as $\tilde{G}$, that overlays G with its dual $\hat{G}$, as shown in Figure 3 (c). Each point in $\tilde{G}$ corresponds to an element in G as well as its dual element in $\hat{G}$. Recall that Dual Contouring proceeds by first creating one vertex for each grid cell that is non-empty (i.e., containing grid points with different signs), followed by creating one polygon for each non-empty grid edge. Iso-surface extraction on G proceeds similarly in two steps:

1) Create one vertex for each pair of a *positive* point and a *negative* cell that contains the point (such pair corresponds to a non-empty cell in $\tilde{G}$).
2) For each pair of a *positive* N-D element σ and a *negative* $(N+1)$-D element δ that contains σ (such pair corresponds to a non-empty edge in $\tilde{G}$), create one polygon connecting vertices created for each pair of a point contained by σ and a cell containing δ.

While Dual Contouring guarantees to produce a crack-free iso-surface, applying the above algorithm in a valid ESO further ensures a manifold output (see proof in Appendix A). To reproduce geometry details, each vertex created in the first step for a point-cell pair is located at the cell's *representative vertex*. The representative vertex of a non-empty octree cell is obtained during ESO construction either from scalar values at grid points or by sampling polygonal geometry (provided by PolyMender [23]). If a representative vertex does not exist (e.g., in a newly created non-empty cell after topology repair), the vertex associated with the point-cell pair is temporarily located halfway between the point and cell centroid. A post-processing step is then applied to smooth these temporary vertices using iterative averaging [25].

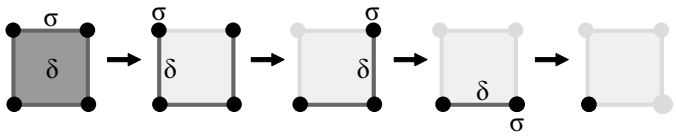


Fig. 4. Thinning using simple removals. Each simple removal (indicated by an arrow) removes a simple element σ and its witness δ. Thinning terminates (far right) when no more simple elements can be found.

Iso-surface extraction can be implemented as tree traversals on the ESO grid. We utilize the recursive procedures proposed in [24], which visits each grid element together with leaf cells sharing the element in one octree traversal. Using these procedures, step (1) is performed in one traversal of all grid points, and step (2) is performed in another traversal of all grid points, edges and faces. Details of the procedures can be found in [24].

V. Handle Removal

Given an input model represented as an ESO grid G, our method removes handles on the iso-surface of G in three steps: thinning, skeleton cycle removal, and growing. Performing these steps on the object V results in cutting ring-like handles, while performing the same steps on the background $\overline{V}$ results in filling tunnel-like handles.

We first describe how each step is performed on the object V, although the same algorithms are equally applied to the background $\overline{V}$. We next show that these steps result in robust removal of existing handles without introducing additional handles. Finally, we discuss efficient implementations of the algorithms on the octree.

A. Algorithms

1) Thinning: The skeleton of the object is obtained by *thinning*, which iteratively removes elements from the object boundary. In order to preserve the topology of the object, thinning is restricted to simple elements:

Definition 1: An N-D element σ in a cellular complex $\mathscr{V}$ is called *simple* with respect to $\mathscr{V}$ if σ is contained in exactly one $(N+1)$-D element δ of $\mathscr{V}$. In particular, δ is called the *witness* of σ.

At each step of thinning, we remove a simple element **together** with its witness from the object. We call the removal of such a pair a *simple removal*. Thinning using simple removals is illustrated in Figure 4, where a simple edge and its witness face are removed first, followed by a sequence of simple removals, each deleting a simple point and its witness edge. Thinning stops when no more simple elements can be found (e.g., a single point is not a simple element based on Definition 1).

2) Skeleton cycle removal: The skeleton generated by thinning may consist of points, edges and faces. Let S_V be the skeleton of V. We consider the *skeleton graph* whose edges are *isolated* edges (i.e., edges with no incident faces) on S_V, denoted as I_{S_V}, and whose nodes are connected components in the remainder $S_V \setminus I_{S_V}$. Note that when the skeleton S_V contains only points and edges, the skeleton graph is S_V itself. Observe in the 2D example of Figure 2 (b) that each cycle in the skeleton graph lies centered in a ring-like handle of V.

Ideally, we would like to identify small handles and to "cut" open a handle ring at its thinnest location. To this end, we shall associate a thickness value at each isolated skeleton edge, which measures the cross-section area of the object at that edge (discussed next). Given the thickness-weighted skeleton graph, we compute the complement of the *maximum* spanning tree (or spanning forrest if $c[S_V] > 1$) of the skeleton graph, and denote E as those edges in this complement whose thickness value falls below a user-specified threshold ε. Removing E from the graph only cuts those cycles whose minimum thickness is smaller than ε, and the cuts (i.e., E) take place at the thinnest portion of each cycle (see Figure 2 (c)). Accordingly, the modified skeleton S'_V is computed as $S'_V = S_V \setminus E$.

a) Generating sets: To explain the thickness measure, we first introduce the *generating set*, $W[e]$, of an isolated edge e in the skeleton S_V. Formally, $W[e] \in V$ is defined as the minimum set so that $V \setminus W[e]$ is a cellular complex, and thinning $V \setminus W[e]$ yields $S_V \setminus \{e\}$. Intuitively, $W[e]$ is a solid "slice" of the object, such that removing the edge e from the skeleton is the same as removing the slice $W[e]$ from the object and applying thinning. Note that generating sets are related to stable manifolds in a flow complex [26]. While the latter relies on a smooth Euclidean distance function, the former is defined by iterative thinning on a discrete grid.

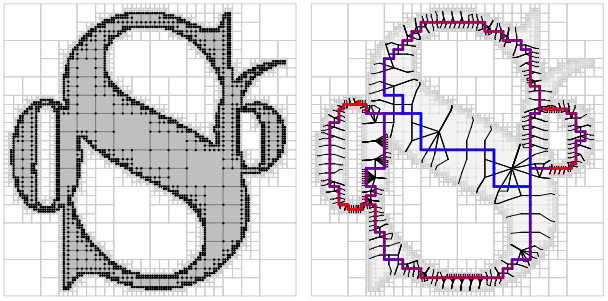
Based on the thinning process that reduces V to S_V, we present a recursive construction for the generating sets:

$$W[\sigma] = \{\sigma\} \cup \bigcup_{\delta \in P[\sigma]} (W[\delta] \cup W[s[\delta]]) \tag{1}$$

where $\sigma \in V$ is any N-D element, $P[\sigma] \in V$ is the set of all $(N+1)$-D elements containing σ, and $s[\delta]$ is the element removed together with δ in a simple removal (i.e., a simple element of which δ is the witness, or the witness of δ) when thinning V to S_V.

To show that equation 1 meets our definition of a generating set, we first observe that $V \setminus W[\sigma]$ is a cellular complex for any σ. This is because any element in V containing an element in $W[\sigma]$ belongs to $W[\sigma]$. In addition, for an isolated edge e, all elements in $W[e]$ but e are paired in simple removals. Hence $V \setminus W[e]$ can be thinned to $S_V \setminus \{e\}$ using the same sequence of simple removals, except those in $W[e]$, that reduce V to S_V. Finally, the construction contains only necessary elements and hence $W[e]$ is minimal.

b) Measuring handles: Observe from equation 1 that the dimensions of elements in the generating set $W[e]$ are no smaller than that of e (i.e., 1). Accordingly, its dual elements



(a) (b)

Fig. 5. (a): The original object V. (b): The skeleton S_V with thickness values (red for thin and blue for thick) and the dual elements of generating sets $\hat{W[e]}$ (black cross-section curves) at each skeleton edge e.

$\hat{W[e]}$ in the dual grid $\hat{G}$ contain only points, edges and faces. As the generating set $W[e]$ forms a solid slice of the object V, its dual $\hat{W[e]}$ forms a cross-section surface of V that "cuts across" the isolated edge e. Figure 5 (b) shows a 2D example, where the dual of each generating set forms a cross-section curve.

The thickness at an isolated edge e, denoted as $w[e]$, is therefore defined as the area of this cross-section surface $\hat{W[e]}$. The construction of $W[e]$ in equation 1 gives a recursive evaluation of $w[e]$:

$$w[e] = A[e] + \sum_{\delta \in P[e]} w[s[\delta]] \tag{2}$$

where $A[e]$ denotes the area of the dual face of e in the dual grid $\hat{G}$ and $w[s[\delta]]$ evaluates to zero if $s[\delta]$ is not an edge. To compute $A[e]$, we triangulate the dual face of e using the midpoint of e when the face is not planar. Figure 5 demonstrates the thickness measure on a skeleton computed from a 2D object. Observe that $w[e]$ adapts well to object thickness at various locations.

3) Growing: The final step "grows" the modified skeleton S'_V back into a new object. Instead of reversing the thinning process, which is a global operation, we take a different, local approach. Let E be the edges removed from the original skeleton S_V, i.e., $E = S_V \setminus S'_V$. We simply *subtract* the generating sets associated with edges in E from the original object V. The new object is thus computed as $V' = V \setminus \bigcup_{e \in E} W[e]$.

4) Cutting and filling handles: The above three steps can be applied to either the object V and the background $\overline{V}$, with the effect of either *cutting* the ring-like handles or *filling* the tunnel-like handles. We illustrate results of cutting and filling using a simple 2-holed torus in Figure 6. Specifically, we let the user specify two different thresholds $\varepsilon, \overline{\varepsilon}$. We first cut rings on V that are thinner than ε, creating a modified object V', and next fill tunnels on V' that are narrower than $\overline{\varepsilon}$. Observe in Figure 6 that due to the use of our thickness measure, each cutting and filling always takes place at the thinnest location of a ring or the narrowest location of a tunnel.

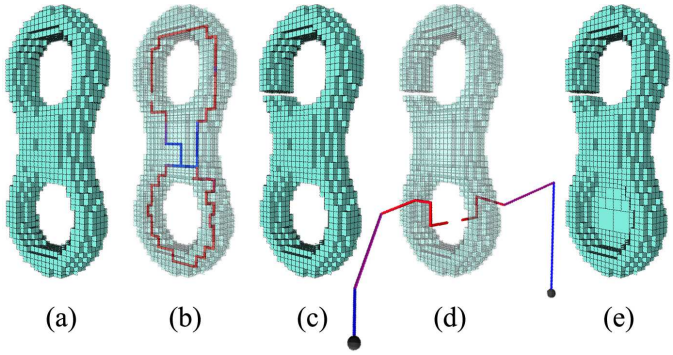


Fig. 6. Removing handles on a 2-holed torus: (a) The original object, (b,c) cutting the top ring, (d,e) filling in the bottom tunnel. Edge thickness on the skeletons are shown from red (small) to blue (big). Black spheres at the ends of the skeleton in (d) are topologically the same point in the dual grid $\hat{G}$ that is dual to the outside cell of the primal grid G.

B. Robustness of handle removal

Let M be the iso-surface on the input ESO grid, and M' be the iso-surface on the modified ESO grid after performing thinning, skeleton cycle removal, and growing. Here we show that M' has exactly m fewer handles than M, where m is the number of cycles removed from the skeleton graph.

Using Euler's formula, the number of handles on a closed manifold iso-surface M is computed by its genus $g[M]$:

$$g[M] = c[M] - \chi[M]/2 \tag{3}$$

where c and χ are the number of connected components and the Euler characteristic. The Euler characteristic of a 3D cellular complex $\mathscr{V}$ is defined as the alternating sum

$$\chi[\mathscr{V}] = k_0[\mathscr{V}] - k_1[\mathscr{V}] + k_2[\mathscr{V}] - k_3[\mathscr{V}],$$

where $k_i[\mathscr{V}]$ enumerates the number of points, edges, faces and cells in $\mathscr{V}$ for $i = 0,1,2,3$ [27] (surface M can be considered as a special cellular complex with no cell elements).

The robustness of our method is built upon the following two equalities that relate the topology of M to that of the object V and the background $\overline{V}$ (see proof in Appendix A):

$$\begin{array}{rcl} c[M] & = & c[V] + c[\overline{V}] - 1 \\ \chi[M] & = & 2\chi[V] = 2\chi[\overline{V}] \end{array} \tag{4}$$

The key observation from equation 3 and equation 4 is that the number of handles on the iso-surface M depends **entirely** on the Euler characteristic and connected components of the object V and background $\overline{V}$, that is:

$$g[M] = c[V] + c[\overline{V}] - 1 - \chi[V] = c[V] + c[\overline{V}] - 1 - \chi[\overline{V}] \tag{5}$$

To confirm our hypothesis that $g[M'] = g[M] - m$, where m is the number of cycles removed from the skeleton graph, we only need to show that the three-step topology repair **increases** $\chi[V]$ (or $\chi[\overline{V}]$) by m while **preserving** both $c[V]$ and $c[\overline{V}]$ (without loss of generality, each step is demonstrated on V):

1) Thinning: A simple removal is in fact equivalent to an *elementary simplicial collapse* in algebraic topology [28], which preserves the homotopy type of a 3-manifold. Let S_V be the skeleton of V after simple removals, we have

$$\chi[S_V] = \chi[V], \quad c[S_V] = c[V], \; c[\overline{S_V}] = c[\overline{V}] \tag{6}$$

2) Skeleton cycle removal: By computing the spanning tree of the graph of S_V, the modified skeleton S'_V preserves the connectivity of S_V while removing as many isolated edges as the cycles removed from the graph:

$$\chi[S'_V] = \chi[S_V] + m, \quad c[S'_V] = c[S_V], \; c[\overline{S'_V}] = c[\overline{S_V}] \tag{7}$$

3) Growing: By definition of generating sets, thinning the new object V' yields the skeleton S'_V. Combining equation 6 and 7, we have

$$\chi[V'] = \chi[V] + m, \; c[V'] = c[V], \; c[\overline{V'}] = c[\overline{V}]. \tag{8}$$

C. Implementation

1) Thinning: Thinning of the object V is performed on an ESO grid in an iterative manner. During each iteration, we make two octree traversals. In the first traversal, we mark every positive point, edge and face that is simple by Definition 1. In the second traversal, we visit each marked element σ and, if σ is still simple at the time of visit, invert the sign of both σ and its witness. The two traversals simulate the peeling of elements on the outmost layer of V. Thinning terminates if no simple elements are found in the first octree traversal. In our implementation, we use the recursive procedures detailed in [24] for efficient traversing of octree grid elements.

Note that thinning of the background $\overline{V}$ can still be performed using octree traversals on the primal grid G based on the following observation: the dual of an N-D negative element δ, denoted as $\hat{\delta}$ in $\hat{G}$, is simple with respect to $\overline{V}$ if δ contains exactly one $(N-1)$-D negative element σ in G.

2) Handle measurement: We compute the thickness measure $w[e]$ for each isolated skeleton edge e during thinning by slightly modifying the two octree traversals described above. Note that a positive face may be the witness of more than one simple edge. To obtain a minimal thickness measure, in the first octree traversal, we associate a face f with the minimal $w[e]$ of all simple edges e that f contains. In the second traversal, we invert the signs of a simple edge e and its witness face f only if $w[e]$ equals the minimal value stored at f, and we update the thickness measure on the remaining edges of f using equation 2.

3) Growing: Growing involves only local modifications of the original object using the generating sets. To construct the generating sets using equation 1, we maintain pointers that track the simple elements from their witnesses during thinning. Note that growing typically takes negligible time due to the small proportion of the handles relative to the entire volume.

VI. Results

We first perform handle removal on a synthetic tree model with genus 18 in Figure 7. Observe that the weighting of

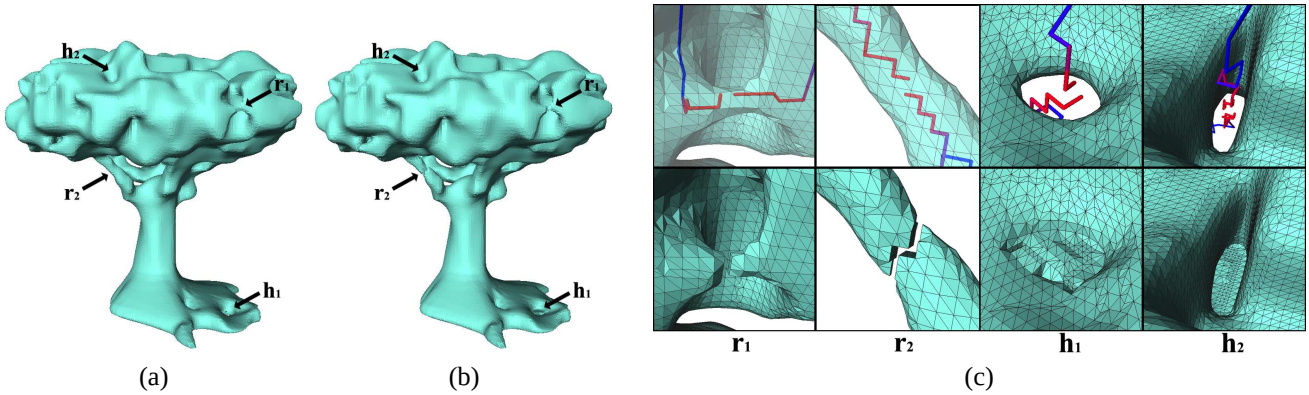


Fig. 7. (a) A tree model with genus 18. (b) The topologically repaired model with genus 0 using cutting threshold $\varepsilon = 0.01$ and filling threshold $\overline{\varepsilon} = 0.04$. (c) Closeup views of the rings (r_1, r_2) and tunnels (h_1, h_2), where the top row shows the original surface with the modified skeleton, and the bottom row shows the modified surface.

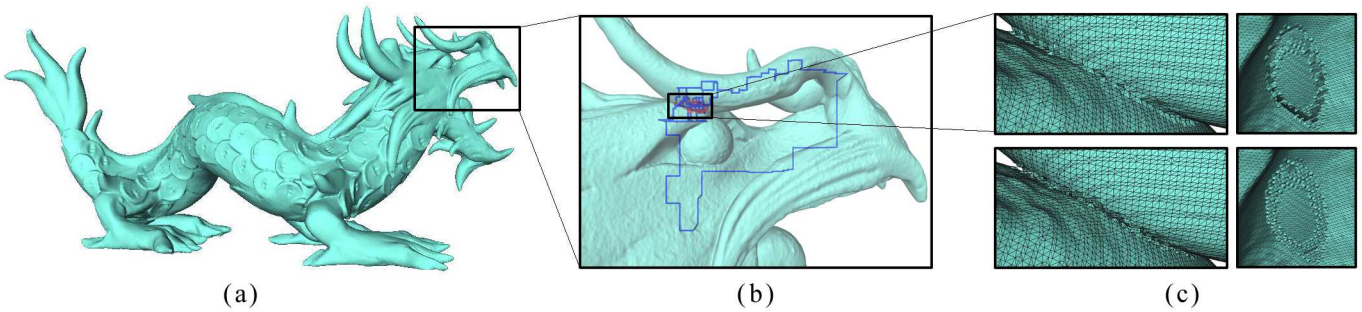


Fig. 8. Topology repair of the Asian Dragon model at octree depth 11. The input model contains several handles where the horn touches the head (b), resulted from geometric repair of the original, self-intersecting polygonal model by PolyMender [23]. Close-up looks at the handles site before and after repair are shown in (c) top and bottom, where the pictures on the right are viewed from inside the dragon head.

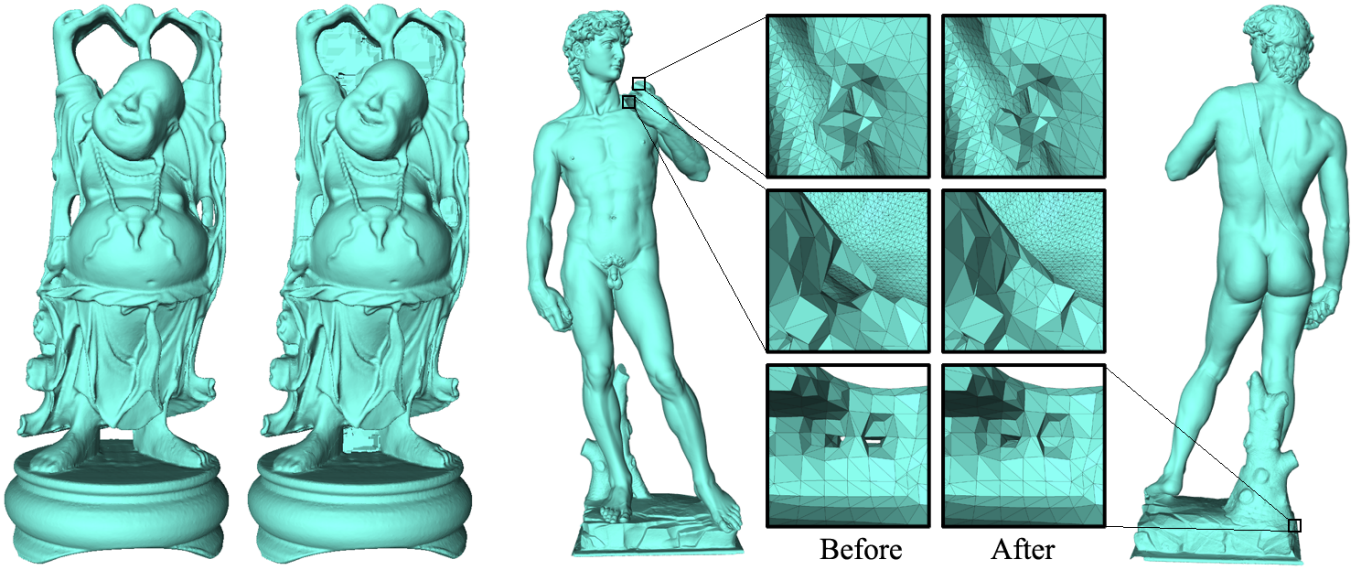


Fig. 9. Topology repair of the Stanford Buddha model at octree depth 10 (left), showing a genus-6 and a genus-0 result, and the 2mm David model repaired at octree depth 12 (right), showing closeup views of the cut and filled handles.

skeleton edges using our thickness measure correctly identifies the thinnest portion of each ring-like handle to be cut and the narrowest portion of each tunnel-like handle to be filled. In addition, handle removals result in only local modifications of the volume, and the geometry away from the modification sites are preserved.

Figure 1 shows how our method differentiates handles of various sizes and removes complex handles in a robust manner. The spider-web model shown on the left is reconstructed from a noisy point cloud and contains 75 handles, many of which are small. Entangling rings and tunnels are shown in the close-up views. By performing filling with an appropriate threshold,

Model	Octree Depth	Octree Leaf Cells	PolyMender Time (sec)	Genus Before	ε	$\overline{\varepsilon}$	Genus After	Cut Time (sec)	Fill Time (sec)	Contour Time (sec)	Total Time (sec)	Memory Usage (MB)	Output Triangles
Spider Web	7	63799	6.8	75	0.0	0.003	17	0	2.5	0.4	2.9	6	65710
Tree	7	181945	2.2	18	0.01	0.04	0	3.5	5.9	1.6	11.0	16	134364
Knotty Mug	8	437802	1.9	2	0.01	0.01	0	6.7	12.0	3.2	21.9	38	378004
Buddha	10	3989252	49.5	11	0.001	0.001	6	62.3	130.1	29.5	221.9	336	3434166
Asian Dragon	11	13978434	264.1	17	0.0005	0.0005	0	218.7	427.3	103.2	749.2	1173	11987648
David (2mm)	12	20749723	330.9	10	0.0005	0.0005	4	325.9	638.0	157.7	1121.6	1743	17815146

TABLE I

PERFORMANCE RESULTS ON PROCESSING VARIOUS MODELS ON A CONSUMER LEVEL PC WITH 3.0GHZ CPU AND 2GB MEMORY. TIMING EXCLUDES I/O DURING CONTOURING.

all handles but the 17 main "holes" of the spider-web are removed, and no additional handles are created.

We demonstrate our method on large scanned models in Figure 8 and 9. The Happy Buddha, Asian Dragon and Michelangelo's David (reconstructed at 2mm resolution) are processed respectively at octree depth 10, 11 and 12, equivalent to a grid of size 1024^3, 2048^3 and 4096^3. To the best of our knowledge, topology repair at the latter two resolutions have not been reported before. Note in particular that the original Asian Dragon mesh from the Stanford 3D Scanning Repository contains a self-intersection where the horn penetrates into the head. Mesh repair using PolyMender results in a number of topological handles at that location (see Figure 8 (b)). Our method removes all handles and separates the horn from the head (see Figure 8 (c)).

Statistics for each model, including the handle thresholds, are reported in Table I. The thresholds $(\varepsilon, \overline{\varepsilon})$ are specified as the ratio of the area of the cross-section surfaces to the area of a side of the ESO bounding box. In each example, the ESO grids are created by first converting from polygonal formats to an octree grid using the PolyMender software [23] (timing is reported). Genus are computed on the iso-surface of the ESO grid. All tests are performed on a 3.0GHz P4 machine with 2G RAM. Note that even on a 4096^3 grid, the entire process finishes in less than twenty minutes on a consumer-level PC.

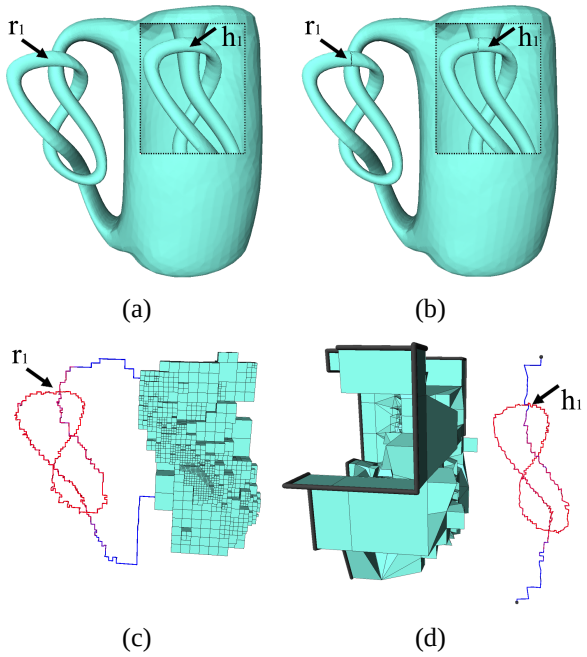


Fig. 10. Removing handles on a mug (a) with both an outside knot r_1 and an inside knot complement h_1, the result is in (b). The object skeleton (c) and the background skeleton (d) each contains surfaces yet captures one of the two handles as isolated skeleton edges. (Thick edges in (d) are topologically identified as the same point on the dual grid dual to the infinite cell on the primal grid).

VII. DISCUSSION

Here we further examine the robustness of our algorithm on solid models with uncommon topologies. In particular, we examine when the skeleton contains faces besides points and edges, and show how a particular type of complex handles is removed with no new handles introduced.

For all of models that we have tested so far, we observed that the skeletons of the object V and the background $\overline{V}$ consist of only points and edges. However, an arbitrary model may contain convoluted features, such as internal cavities, complements of 3D knots and the "house-with-two-rooms" [27], which yield skeletons containing faces that form closed surfaces. Figure 10 (a) shows an extreme case where a two-handled mug has a knotted handle on the outside and a knot complement on the inside. As a result, the object skeleton S_V contains faces around the knot complement while the background skeleton $S_{\overline{V}}$ contains faces around the knotted handle, as shown in (c,d). Nevertheless, the skeleton graphs still detect the handles as graph cycles, because each handle reduces to isolated skeleton edges in **either** the object skeleton S_V **or** the background skeleton $S_{\overline{V}}$. Combining cutting and filling, the two handles are removed, as shown in (b).[2]

For each handle detected as a cycle on the skeleton graph, our method guarantees removal of the handle without introducing new handles. We especially demonstrate this advantage in removing a *blocked* handle, as shown in Figure 11. The genus-2 torus in (a) contains a tunnel inside (as shown in wireframe in (c)), which connects to the outside through an outlet at the top. Note that simply cutting the torus ring at an arbitrary location will introduce a new handle (i.e., the total genus remains 2) due to the presence of the tunnel that "blocks" the cut. Our method results in filling of the interior tunnel (highlighted in (d)) while cutting the torus ring at the tunnel outlet, which yields a genus-0 output.

[2]Although lacking formal proof, we hypothesize that any surface handle can be detected using the skeleton graph of either S_V or $S_{\overline{V}}$.

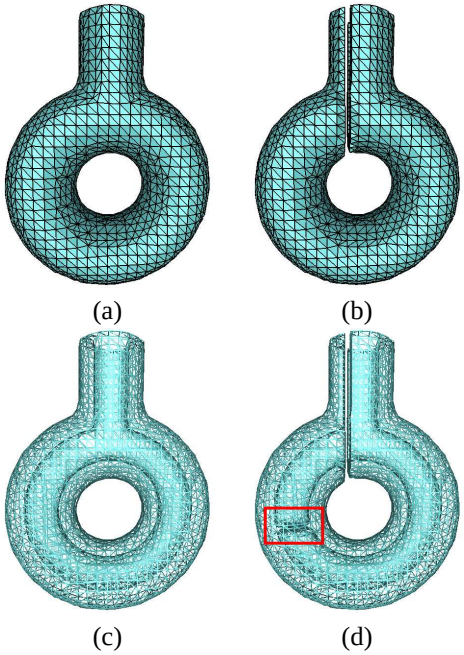


Fig. 11. Removing a blocked handle. (a,c): A genus-2 torus containing a tunnel inside (with an outlet at the top). (b,d): Handles removed by filling the interior tunnel (see red square) and breaking the exterior torus. Note that no new handles are introduced.

VIII. Conclusion

We present a novel volumetric method for removing topological errors on solid models in the form of small handles resulted from surface reconstruction. Our method is based on computing a skeleton representation using morphological operations on an adaptive grid structure. For each handle removed, either by cutting the ring or by filling the tunnel, our method guarantees not to introduce additional handles. In addition, large models can be processed at very high resolutions in an efficient manner.

Our current method has several limitations, and we are investigating possible solutions as part of our future research. First, just as other volumetric mesh-repair methods [16], [17], [23], our approach requires the entire input model to be converted to and from a volume grid, and loss of geometric details may occur when the input is in polygonal format (despite the fact we use vertices directly sampled from the original geometry, see Section IV.C). A possible extension is to apply our volumetric repair only to portions of a mesh that have been identified to contain topology errors using mesh-based approaches. This hybrid idea has already been realized in a different setting for repairing geometric errors on CAD models [30].

Second, like previously proposed measures of handle size based on surface area [15] and geodesic loop lengths [17], our cross-section-area-based measure is not always indicative of a feature handle versus a topological error. Similarly, cutting and filling may not be the best way to resolve handles in all cases. The right measure and removal scheme should consider the source of topological errors, which varies by how the input model was created (e.g., from scanned data, using CAD software, from medical images, etc.). Although we do not assume a particular source of topological errors in this paper, it will be interesting to examine new measures that cater to specific types of input models, and new ways for resolving handles (e.g., approximating small entangling handles on an otherwise smooth surface using a single smooth patch).

Moreover, we will investigate improved thinning methods that extend recent level-set techniques [29] on uniform grid to ensure a uniform thinning speed on adaptive grids, which will yield a smoother skeleton as well as handle cuts with less bias towards axes directions. Such thinning techniques will be useful in general for extracting shape-preserving skeletons of large models.

Acknowledgment

The authors would like to thank Stanford University Computer Graphics Laboratory for providing the scanned models (Buddha, Dragon and David) and Cindy Grimm for providing the Tree and Spider-web models. We would also like to thank Cindy Grimm and Rachel Roberts for their helpful discussions.

References

[1] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Forms*, W. Wathen-Dunn, Ed. Amsterdam: MIT Press, 1967, pp. 362–380.

[2] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*. New York, NY, USA: ACM Press, 2001, pp. 249–266.

[3] T. K. Dey and W. Zhao, "Approximate medial axis as a voronoi subcomplex," in *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications*. New York, NY, USA: ACM Press, 2002, pp. 356–366.

[4] I. Ragnemalm, "The euclidean distance transformation in arbitrary dimensions," *Pattern Recognition*, vol. 14, pp. 883–888, 1993.

[5] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker, "The hamilton-jacobi skeleton," in *International Conference on Computer Vision (ICCV)*, 1999, pp. 828–834. [Online]. Available: citeseer.ist.psu.edu/article/siddiqi99hamiltonjacobi.html

[6] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, 1992.

[7] K. Palágyi and A. Kuba, "A parallel 3d 12-subiteration thinning algorithm," *Graph. Models Image Process.*, vol. 61, no. 4, pp. 199–221, 1999.

[8] G. Bertrand, "Simple points, topological numbers and geodesic neighborhoods in cubic grids," *Pattern Recogn. Lett.*, vol. 15, no. 10, pp. 1003–1011, 1994.

[9] 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," *Journal of Mathematical Imaging and Vision*, vol. 5, pp. 297–318, 1995.

[10] Z. Aktouf, G. Bertrand, and L. Perroton, "A 3d-hole closing algorithm," in *6th International Workshop on Discrete Geometry for Computer Imagery*, 1996, pp. 36–47.

[11] N. Kriegeskorte and R. Goebel, "An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical mr volumes." *Neuroimage*, vol. 14, no. 2, pp. 329–346, August 2001.

[12] S. Bischoff and L. Kobbelt, "Isosurface reconstruction with topology control." in *Pacific Conference on Computer Graphics and Applications*, 2002, pp. 246–255.

[13] B. Fischl, A. K. Liu, and A. M. Dale, "Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex." *IEEE Trans. Med. Imaging*, vol. 20, no. 1, pp. 70–80, 2001.

[14] J. El-Sana and A. Varshney, "Controlled simplification of genus for polygonal models." in *IEEE Visualization*, 1997, pp. 403–412.

[15] I. Guskov and Z. J. Wood, "Topological noise removal." in *Graphics Interface*, 2001, pp. 19–26.

[16] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques." *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 2, pp. 191–205, 2003.

[17] Z. J. Wood, H. Hoppe, M. Desbrun, and P. Schröder, "Removing excess topology from isosurfaces." *ACM Trans. Graph.*, vol. 23, no. 2, pp. 190–208, 2004.

[18] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1987, pp. 163–169.

[19] D. W. Shattuck and R. M. Leahy, "Automated graph based analysis and correction of cortical volume topology." *IEEE Trans. Med. Imaging*, vol. 20, no. 11, pp. 1167–1177, 2001.

[20] X. Han, C. Xu, U. Braga-Neto, and J. L. Prince, "Topology correction in brain cortex segmentation using a multi-scale, graph-based algorithm." *IEEE Trans. Med. Imaging*, vol. 21, no. 2, pp. 109–121, 2002.

[21] A. Szymczak and J. Vanderhyde, "Extraction of topologically simple isosurfaces from volume datasets," in *IEEE Visualization*, 2003, pp. 67–74.

[22] S. Bischoff and L. Kobbelt, "Sub-voxel topology control for level-set surfaces." *Comput. Graph. Forum*, vol. 22, no. 3, pp. 273–280, 2003.

[23] T. Ju, "Robust repair of polygonal models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 888–895, 2004.

[24] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 2002, pp. 339–346.

[25] G. Taubin, "A signal processing approach to fair surface design," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1995, pp. 351–358.

[26] J. Giesen and M. John, "The flow complex: a data structure for geometric modeling." in *SODA*, 2003, pp. 285–294.

[27] A. Hatcher, *Algebraic Topology*. Cambridge University Pres, 2002.

[28] S. Matveev, *Algorithmic Topology and Classification of 3-Manifolds*. Springer-Verlag Berlin, 2003.

[29] S. Bischoff and L. Kobbelt, "Topologically correct extraction of the cortical surface of a brain using level-set methods." in *Bildverarbeitung für die Medizin*, 2004, pp. 50–54.

[30] ——, "Structure preserving cad model repair," *Computer Graphics Forum*, vol. 24, no. 3, pp. 527–536, 2005.

APPENDIX I
TOPOLOGY PROPERTY OF ESO ISO-SURFACE

Proposition 1: Let M denote the iso-surface on a valid ESO grid with object V and background $\overline{V}$. Then M is a crack-free, 2-manifold surface satisfying equation 4.

Proof:

1) Crack-free surface: Applying Dual Contouring, each edge (or face) on the iso-surface is dual to a non-empty face (or edge) in the composite grid $\tilde{G}$. Since each non-empty grid face always contains an even number of non-empty grid edges, each edge on the iso-surface is shared by an even number of faces, and the surface is closed.

2) Manifold surface: Consider a non-empty face $\tilde{f}$ in the composite grid $\tilde{G}$. When G is valid, an element σ in G (or dual grid $\hat{G}$) must be positive (or negative) if some element containing σ is positive (or negative). As a result, positive points and negative points in $\tilde{f}$ always form two edge-connected components. By duality, the iso-surface edge dual to $\tilde{f}$ is shared by two polygons. Similarly, we can show that the positive points and negative points in a non-empty cell in the composite grid $\tilde{G}$ always form two connected components, and hence the iso-surface vertex dual to the cell is contained in a manifold neighborhood.

3) $\chi[V] = \chi[\overline{V}]$: Since each N-D element in the $\hat{G} \setminus \overline{V}$ is dual to an $(3-N)$-D element in V, we have $\chi[\overline{V}] - \chi[V] = \chi[\hat{G}]$. On the other hand, observe that $\hat{G}$ is constructed by gluing the interior elements to a single outside point, which topologically forms a genus-0 surface in 4D. Hence we have $\chi[\overline{V}] - \chi[V] = \chi[\hat{G}] = 0$.

4) $\chi[M] = \chi[V] + \chi[\overline{V}]$: Consider the decomposition of $\tilde{G}$ into non-empty elements ($\tilde{M}$), elements containing only positive points ($\tilde{V}$) and elements containing only negative points ($\tilde{\overline{V}}$). Note that $\chi[V] = \chi[\tilde{V}]$ and $\chi[\overline{V}] = \chi[\tilde{\overline{V}}]$. Using Dual Contouring, each N-D element on M is due to a non-empty $(3-N)$-D element in $\tilde{G}$, hence $\chi[M] = -\chi[\tilde{M}]$. For the same reason that $\chi[\hat{G}] = 0$, we have $\chi[V] + \chi[\overline{V}] - \chi[M] = \chi[\tilde{G}] = 0$.

5) $c[M] = c[V] + c[\overline{V}] - 1$: The connected components of V and $\overline{V}$ can be represented as nodes in a connected, acyclic graph where each edge denotes a connected piece of surface separating an object component and a background component. The equality therefore holds by graph theory.

□