Topology Repair of Solid Models Using Skeletons

Qian-Yi Zhou, Tao Ju, and 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, thinning, octree.

1 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 watertight surfaces from these data, the resulting models may still exhibit topological errors in the form of small handles such as those shown in Fig. 1a. 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 due either to operations that require the full mesh resolution or to 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 converts an input solid into a volume grid and thins the volumetric model to a skeleton so that the task of detecting handles is reduced to identifying cycles on 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 the 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. Although 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 (for example, $4,096^3$) in minutes.

2 Related Work

2.1 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 in [6] and a 3D introduction in [7]).

- *Q.-Y. Zhou and S.-M. Hu are with the Graphics and Geometric Computing Group, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China. E-mail: zqy@mails.tsinghua.edu.cn, shimin@tsinghua.edu.cn.*
- *T. Ju is with the Department of Computer Science and Engineering, Washington University in St. Louis, One Brookings Drive, Campus Box 1045, St. Louis, MO 63130. E-mail: taoju@cs.wustl.edu.*

Manuscript received 10 May 2006; revised 1 Dec. 2006; accepted 26 Dec. 2006; published online 22 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0066-0506.
Digital Object Identifier no. 10.1109/TVCG.2007.1015.

1077-2626/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

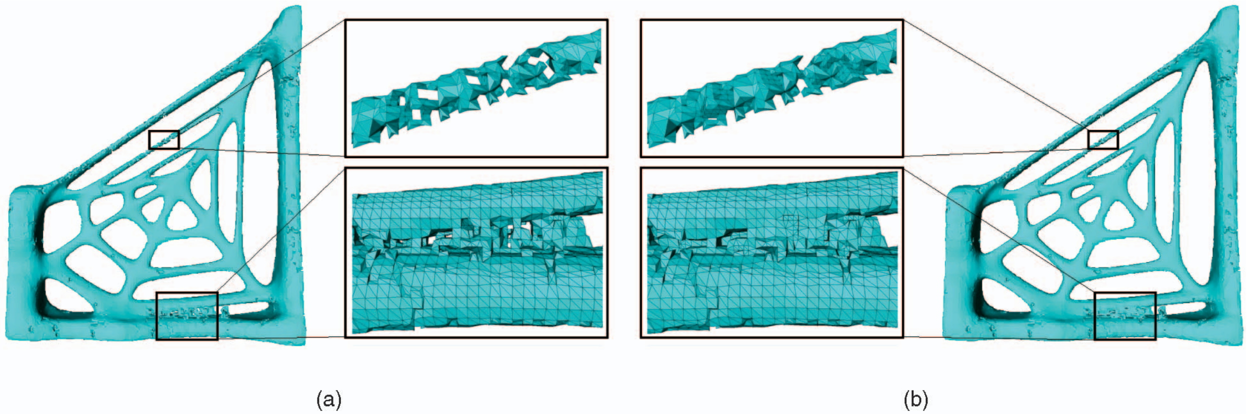


Fig. 1. Topology repair on a spiderweb model: (a) The original genus-75 model reconstructed from point sets with many small and entangled handles. (b) Topology repair removes all erroneous handles except for the 17 major Web holes.

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.

2.2 Topology-Controlled Surface Reconstruction

This class of methods [9], [10], [11], [12] is designed to reconstruct isosurfaces 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. Although such methods have been effective in reconstructing topological spherical cortical surfaces from magnetic resonance imaging (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.

2.3 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 manner, Guskov and Wood [15] employ 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], and by exploring a surface neighborhood that can be potentially large for identifying long and thin handles [15]. Note that optimizations can be performed to dramatically speed up the detection of small handles, as proposed in a recent work by Attene and Falcidieno [16].

2.4 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 [17] 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. [18] detects each surface handle as a cycle in the Reeb graph of the isosurface extracted using the Marching Cubes method [19] and performs handle removal by filling a disklike volume inside the shortest geodesic loop corresponding to each cycle. Still, the main problem with this hybrid approach, as commented by the authors, is the possible introduction of new handles due to the loop-filling operation. In addition, the removal of each handle requires rebuilding the Reeb graph on a uniform grid, which can be time consuming for a large number of handles. The detection of the 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 [20] and Han et al. [21]. 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 [20] is based on axes-aligned sweeping, whereas handle removal using the graph in [21] requires nontrivial 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.

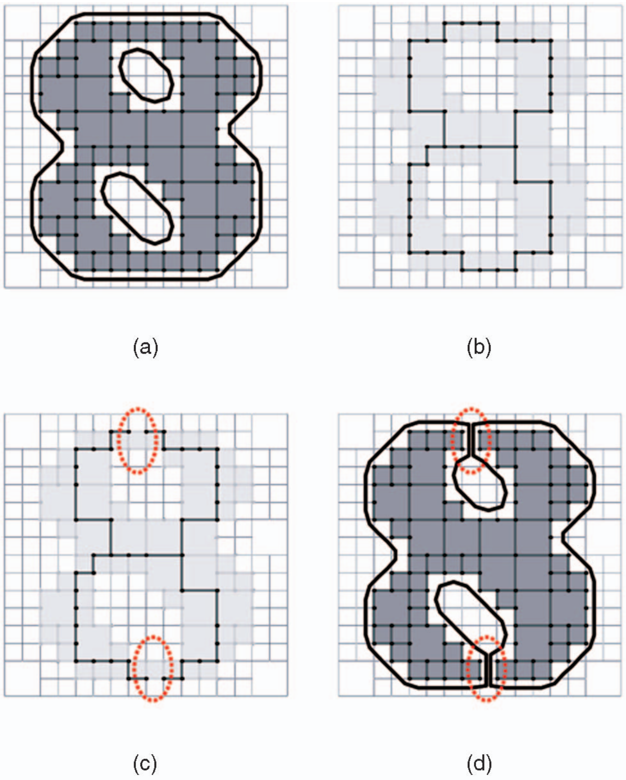


Fig. 2. Two-dimensional illustration of handle removal using skeletons. (a) The original object (darkened points, edges, and faces) and the isosurface (solid lines). (b) The skeleton of the object. (c) The modified skeleton consisting of a spanning tree of (b) (removed edges are highlighted). (d) The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

Recently, a multiresolution solution was proposed by Szymczak and Vanderhyde [22], which applied topology-preserving carving operations to extract isosurfaces 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 Fig. 6e).

3 Method Overview

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

1. **Thin** the object into a *skeleton* that preserves the topology of the object (Fig. 2b).
2. **Remove** cycles in the skeleton by computing the spanning tree of the graph defined by the skeleton (Fig. 2c).
3. **Grow** the modified skeleton to form a new object that preserves the topology of the skeleton (Fig. 2d).

Intuitively, a cycle in the skeleton corresponds to a ringlike handle on the original surface, and removing the skeleton cycle has the effect of "cutting" the ring at the location where the cycle is cut (see Figs. 2c and 2d). In our method, removing one skeleton cycle is guaranteed to cut exactly one surface handle without introducing additional handles (unlike mesh-based [13], [14], [15] or previous volumetric [17], [18] 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.

4 Volume Representation

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

4.1 Motivation

Our volume representation is motivated by the need to analyze and modify the topology of a solid using its skeleton. As we shall see in Section 5, our topology analysis and operations require models, as well as their skeletons, to be represented as 3D *cellular complexes*, which consist of points (0D), edges (1D), faces (2D), and cells (3D). In particular, each edge connects two points, each face is enclosed by a ring of edges, and each cell is enclosed by an envelope of faces. For example, the darkened points, edges, and faces in each grid in Fig. 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 (for example, 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 the highlighted region in Fig. 2c). Even the more advanced representation [23] restricts tagging to just points and edges.

4.2 Representation

We begin with an octree structure to support the 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 (for example, 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

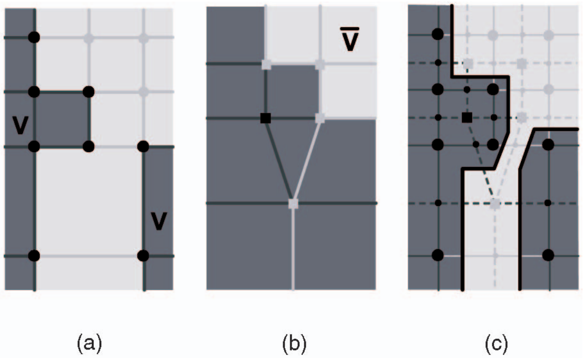


Fig. 3. (a) A primal ESO grid G. (b) The dual grid $\hat{G}$. (c) The composite grid $\tilde{G}$ constructed by overlaying G with $\hat{G}$ and the isosurface (solid lines). In (a) and (b), positive grid elements are darkened and negative elements are dimmed.

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 an ESO grid and its dual are shown in Figs. 3a and 3b. 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 Fig. 3b). 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 the 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 $\overline{\overline{V}} = V$.

4.3 Operations

4.3.1 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 (for example, MRI data) or from a polygonal mesh using scan-conversion routines. In this paper, we use the PolyMender software [24], which is capable of producing a watertight solid model from arbitrary polygonal soups.

4.3.2 Extracting the Isosurface

To construct the isosurface of an ESO grid G, we extend the Dual Contouring algorithm [25], which was designed for octrees with signs only 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 Fig. 3c. 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 nonempty (that is, containing grid points with different signs), followed by creating one polygon for each nonempty grid edge. The isosurface 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 a pair corresponds to a nonempty 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 a pair corresponds to a nonempty edge in $\tilde{G}$), create one polygon connecting vertices created for each pair of a point contained by σ and a cell containing δ.

Although Dual Contouring guarantees producing a crack-free isosurface, applying the above algorithm in a valid ESO further ensures a manifold output (see proof in the Appendix). 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 nonempty octree cell is obtained during an ESO construction either from scalar values at grid points or by sampling polygonal geometry (provided by PolyMender [24]). If a representative vertex does not exist (for example, in a newly created nonempty cell after topology repair), the vertex associated with the point-cell pair is temporarily located halfway between the point and the cell centroid. A postprocessing step is then applied to smooth these temporary vertices using iterative averaging [26].

Isosurface extraction can be implemented as tree traversals on the ESO grid. We utilize the recursive procedures proposed in [25], which visit 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 [25].

5 Handle Removal

Given an input model represented as an ESO grid G, our method removes handles on the isosurface of G in three steps: thinning, skeleton cycle removal, and growing. Performing these steps on the object V results in cutting ringlike handles, whereas 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 the robust removal of existing handles without introducing additional handles. Finally, we discuss efficient implementations of the algorithms on the octree.

[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.

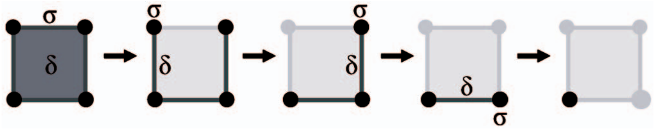


Fig. 4. Thinning using simple removals. Each simple removal (indicated by an arrow) removes a simple element σ and its witness δ. Thinning terminates (far right) when no more simple elements can be found.

5.1 Algorithms

5.1.1 Thinning

The skeleton of the object is obtained by *thinning*, which iteratively removes elements from the object boundary. To carry out topology operations on the resulting skeleton in subsequent steps, thinning should yield cellular complexes that preserve the topology of the original object. In addition, the thinning procedure should admit efficient implementation for processing large models.

To this end, we first introduce the concept of simple elements and their witnesses.

Definition 1. *An N-D element σ in a cellular complex $\mathcal{V}$ is called* simple *with respect to $\mathcal{V}$ if σ is contained in exactly one $(N+1)$-D element δ of $\mathcal{V}$. 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*. The thinning using simple removals is illustrated in Fig. 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. The thinning stops when no more simple elements can be found (for example, a single point is not a simple element based on Definition 1).

Note that, after each simple removal, the remainder of the object is still a valid cellular complex. In addition, performing a simple removal requires only counting the number of $(N+1)$-D elements sharing an N-D element, which can be implemented efficiently using recursive walks on the octree grid (see Section 5.3).

5.1.2 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 (that is, 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 in Fig. 2b that each cycle in the skeleton graph lies centered in a ringlike 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 forest 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 (that is, E) take place at the thinnest portion of each cycle (see Fig. 2c). Accordingly, the modified skeleton S'_V is computed as $S'_V = S_V \setminus E$.

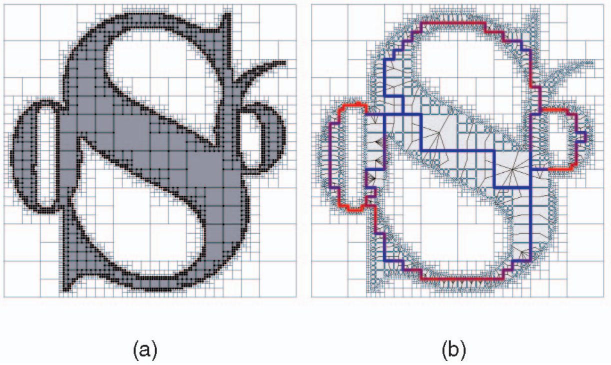


Fig. 5. (a) The original object V. (b) The skeleton S_V with thickness values (red for thin and blue for thick) and the dual elements of generating sets $\hat{W[e]}$ (black cross-section curves) at each skeleton edge e.

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 the generating sets are related to stable manifolds in a flow complex [27]. Although the latter relies on a smooth euclidean distance function, the former is defined by an iterative thinning on a discrete grid.

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

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

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 (that is, a simple element of which δ is the witness or the witness of δ) when thinning V to S_V.

To show that (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 the necessary elements and, hence, $W[e]$ is minimal.

Measuring Handles. Observe from (1) that the dimensions of elements in the generating set $W[e]$ are not smaller than that of e (that is, 1). Accordingly, its dual elements $\hat{W[e]}$ 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. Fig. 5b shows a 2D example, where the dual of each generating set forms a cross-section curve.

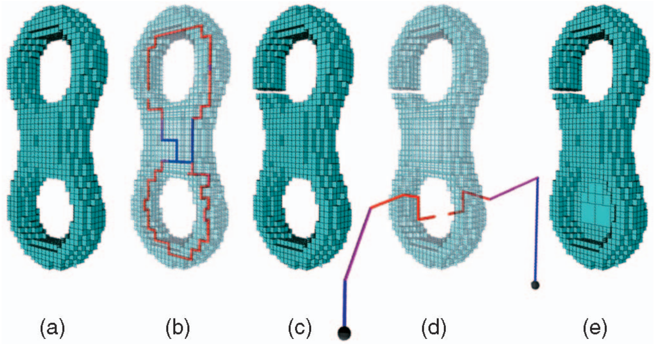


Fig. 6. Removing handles on a two-holed torus. (a) The original object. (b) and (c) Cutting the top ring. (d) and (e) Filling in the bottom tunnel. Edge thickness on the skeletons is shown from red (small) to blue (big). Black spheres at the ends of the skeleton in (d) are topologically the same point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

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

$$w[e] = A[e] + \sum_{\delta \in P[e]} w[s[\delta]], \qquad (2)$$

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. Fig. 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.

5.1.3 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, that is, $E = S_V \setminus S'_V$. We simply *subtract* the generating sets associated with edges in E from the original object V. The new object is thus computed as $V' = V \setminus \bigcup_{e \in E} W[e]$.

5.1.4 Cutting and Filling Handles

The above three steps can be applied to either the object V or the background $\overline{V}$, with the effect of either *cutting* the ringlike handles or *filling* the tunnel-like handles. We illustrate results of cutting and filling using a simple two-holed torus in Fig. 6. Specifically, we let the user specify two different thresholds ε and $\overline{\varepsilon}$. 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 Fig. 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.

5.2 Robustness of Handle Removal

Let M be the isosurface on the input ESO grid and M' be the isosurface 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 isosurface M is computed by its genus $g[M]$:

$$g[M] = c[M] - \chi[M]/2, \qquad (3)$$

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

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

where $k_i[\mathcal{V}]$ enumerates the number of points, edges, faces, and cells in $\mathcal{V}$ for $i = 0, 1, 2, 3$ [28] (surface M can be considered as a special cellular complex with no cell elements).

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

$$\begin{aligned} c[M] &= c[V] + c[\overline{V}] - 1, \\ \chi[M] &= 2\chi[V] = 2\chi[\overline{V}]. \end{aligned} \qquad (4)$$

The key observation from (3) and (4) is that the number of handles on the isosurface M depends **entirely** on the Euler characteristic and connected components of the object V and background $\overline{V}$, that is,

$$g[M] = c[V] + c[\overline{V}] - 1 - \chi[V] = c[V] + c[\overline{V}] - 1 - \chi[\overline{V}]. \quad (5)$$

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 [29], which preserves the homotopy type of a 3-manifold. Letting S_V be the skeleton of V after simple removals, we have

$$\chi[S_V] = \chi[V], \quad c[S_V] = c[V], \quad c[\overline{S_V}] = c[\overline{V}]. \qquad (6)$$

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:

$$\chi[S'_V] = \chi[S_V] + m, \quad c[S'_V] = c[S_V], \quad c[\overline{S'_V}] = c[\overline{S_V}]. \qquad (7)$$

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

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

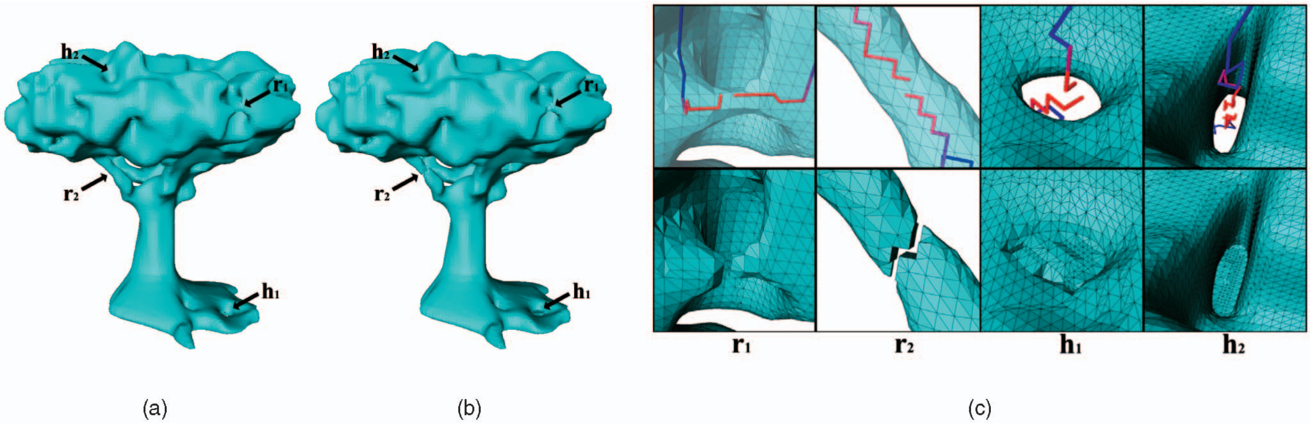


Fig. 7. (a) A tree model with genus 18. (b) The topologically repaired model with genus 0 using cutting threshold $\varepsilon = 0.01$ and filling threshold $\overline{\varepsilon} = 0.04$. (c) Close-up views of the rings (r_1, r_2) and tunnels (h_1, h_2), where the top row shows the original surface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

5.3 Implementation

5.3.1 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 [25] 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.

5.3.2 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 (2).

5.3.3 Growing

Growing involves only local modifications of the original object using the generating sets. To construct the generating sets using (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.

6 Results

We first perform handle removal on a synthetic tree model with genus 18 in Fig. 7. Observe that the weighting of skeleton edges using our thickness measure correctly identifies the thinnest portion of each ringlike 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 is preserved.

Fig. 1 shows how our method differentiates handles of various sizes and removes complex handles in a robust manner. The spiderweb 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, all handles but the 17 main "holes" of the spiderweb are removed, and no additional handles are created.

We demonstrate our method on large scanned models in Figs. 8 and 9. The Happy Buddha, Asian Dragon, and Michelangelo's David (reconstructed at 2 mm resolution) are processed, respectively, at octree depth 10, 11, and 12, equivalent to a grid of size $1{,}024^3$, $2{,}048^3$, and $4{,}096^3$. To the best of our knowledge, the topology repairs 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 Fig. 8b). Our method removes all handles and separates the horn from the head (see Fig. 8c).

Statistics for each model, including the handle thresholds, are reported in Table 1. 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 [24] (timing is reported). Genus is computed on the isosurface of the ESO grid. All tests are performed on a 3.0 GHz P4 machine with a 2 Gbyte RAM. Note that, even on a $4{,}096^3$ grid, the entire process finishes in less than 20 minutes on a consumer-level PC.

7 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

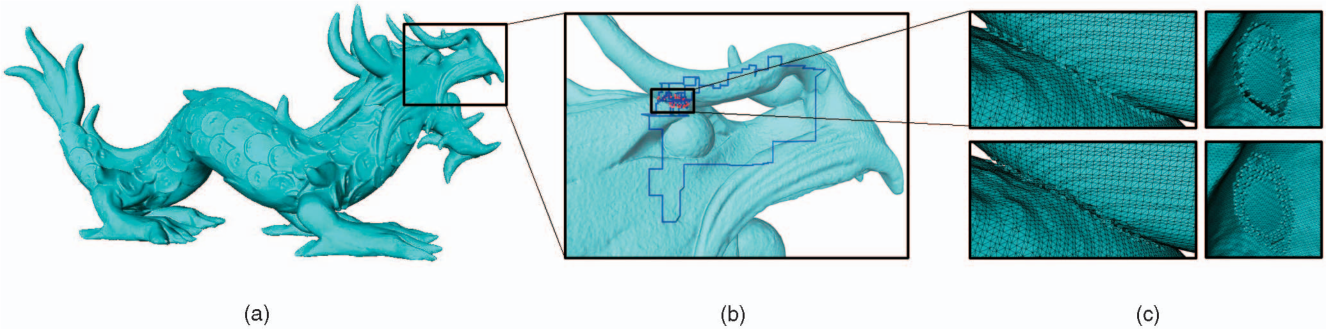


Fig. 8. Topology repair of the Asian Dragon model at octree depth 11. The input model contains several handles where the horn touches the head (b), resulting from the geometric repair of the original self-intersecting polygonal model by PolyMender [24]. Close-up looks at the handles site before and after repair are shown in (c), top and bottom, where the pictures on the right are viewed from inside the dragon head.

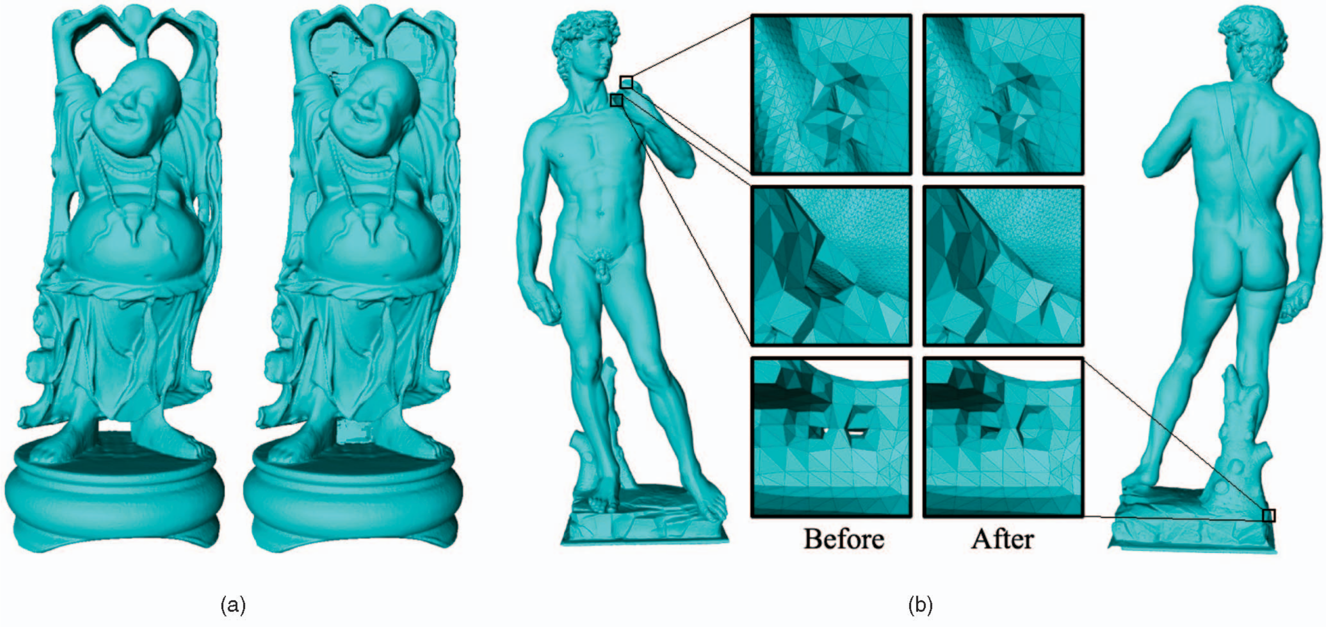


Fig. 9. Topology repair of (a) the Stanford Buddha model at octree depth 10, showing a genus-6 and a genus-0 result, and (b) the 2 mm David model repaired at octree depth 12, showing close-up views of the cut and filled handles.

TABLE 1

Performance Results on Processing Various Models on a Consumer-Level PC with a 3.0 GHz CPU and a 2 Gbyte Memory

Model	Octree Depth	Octree Leaf Cells	PolyMender Time (sec)	Genus Before	ε	$\overline{\varepsilon}$	Genus After	Cut (sec)	Fill (sec)	Contour (sec)	Total (sec)	Memory (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

Timing excludes I/O during contouring.

and edges and show how a particular type of complex handle is removed with no new handle being introduced.

For all 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" [28], which yield skeletons containing faces that form closed surfaces. Fig. 10a 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, whereas the background skeleton $S_{\overline{V}}$ contains faces around the knotted handle, as shown in Figs. 10c and 10d. 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 Fig. 10b.[2]

For each handle detected as a cycle on the skeleton graph, our method guarantees the removal of the handle

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

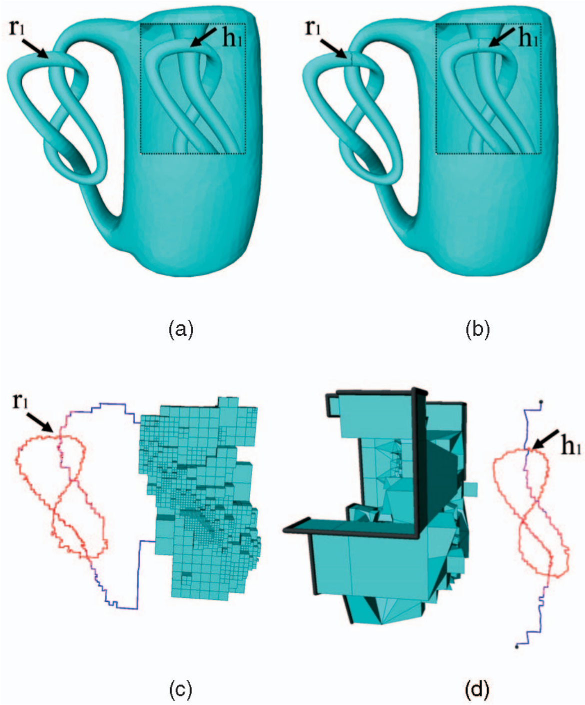


Fig. 10. Removing handles on (a) a mug with both an outside knot r_1 and an inside knot complement h_1; the result is shown in (b). (c) The object skeleton and (d) the background skeleton each contain surfaces and yet capture one of the two handles as isolated skeleton edges. (The thick edges in (d)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

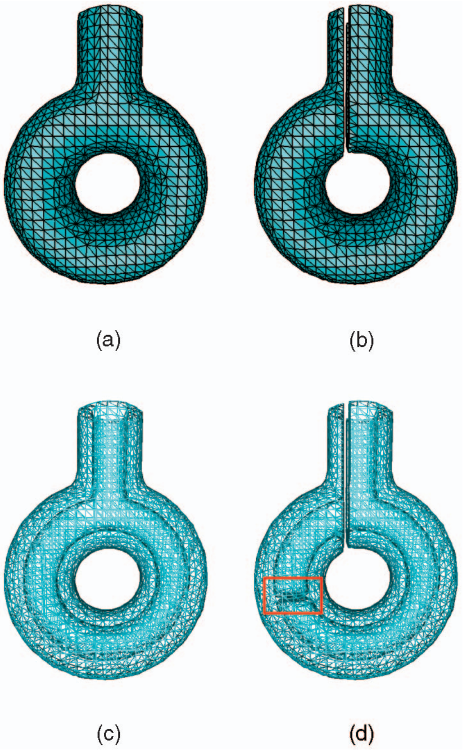


Fig. 11. Removing a blocked handle. (a) and (c) A genus-2 torus containing a tunnel inside (with an outlet at the top). (b) and (d) Handles removed by filling the interior tunnel (see red square) and breaking the exterior torus. Note that no new handles are introduced.

without introducing new handles. We especially demonstrate this advantage in removing a *blocked* handle, as shown in Fig. 11. The genus-2 torus in Fig. 11a contains a tunnel inside (as shown in wireframe in Fig. 11c), 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 (that is, the total genus remains two) due to the presence of the tunnel that “blocks” the cut. Our method results in filling of the interior tunnel (highlighted in Fig. 11d) while cutting the torus ring at the tunnel outlet, which yields a genus-0 output.

8 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. Despite its advantage over mesh-based methods in efficiency and robustness, our method requires volumetric conversion from input models represented in polygonal form. In this paper, we used feature-preserving scan-conversion [24] and isosurfacing routines [25] capable of accurately reproducing geometric features (for example, sharp edges and corners) and utilized an octree grid structure for adaptive representation of high-resolution polygonal geometry. Nonetheless, the originally mesh connectivity and fine geometric details may still be lost in the process. 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].

The presented handle removal method requires two passes over the volume, one for cutting handle rings and one for filling handle tunnels. Although, in each pass, a handle is always cut at its thinnest place or filled at its narrowest location, the removal operation may still modify a larger volume than necessary if cutting is performed where filling would have resulted in a smaller modification, or vice versa. To this end, we are currently investigating the relations between the object skeleton and background skeleton in order to identify the set of edge removals on both skeletons simultaneously that would result in minimum total modification to the volume.

Another research direction that we are currently pursuing is the development of a user interface for interactive topology editing. Note that, in nearly all topology repair methods (including ours), the decisions regarding what handles are to be removed, where the removal takes place along the handle, and how the handle is removed (for example, cutting or filling) are based purely on some heuristic measure of handle sizes computed from the model. Examples of such measures include surface areas in [15], geodesic loop lengths in [18], and cross-section areas in this paper. Given different input data, however, any such measure could possibly fail. In these situations, human judgment is often the best (if not only) criteria. We note that a distinctive advantage of our method over previous work for user interaction is that the skeleton provides a visual form for examining and editing topology. It is not hard to imagine a user interface in which both object and background skeletons are displayed and users identify handles, as well as their exact location and means for their removal conveniently via selecting and removing edges on either skeleton. Such user interface can also provide new means for resolving handles besides cutting a thin opening or filling a thin membrane such as removing an extended segment of the handle (when a user selects a sequence of skeleton edges for removal) and approximating small entangling handles on an otherwise smooth surface using a single smooth patch.

Finally, we will investigate improved thinning methods that extend recent level-set techniques [31] on a 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 toward axes directions. Such thinning techniques will be useful in general for extracting shape-preserving skeletons of large models.

Appendix

Topology Property of ESO Isosurface

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

Proof.

1. *Crack-Free Surface*. Applying Dual Contouring, each edge (or face) on the isosurface is dual to a nonempty face (or edge) in the composite grid $\tilde{G}$. Since each nonempty grid face always contains an even number of nonempty grid edges, each edge on the isosurface is shared by an even number of faces, and the surface is closed.
2. *Manifold Surface*. Consider a nonempty 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 isosurface edge dual to $\tilde{f}$ is shared by two polygons. Similarly, we can show that the positive points and negative points in a nonempty cell in the composite grid $\tilde{G}$ always form two connected components and, hence, the isosurface 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 a $(3-N)$-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 nonempty 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 nonempty $(3-N)$-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. □

Acknowledgments

The authors would like to thank the Stanford University Computer Graphics Laboratory for providing the scanned models (Buddha, Dragon, and David) and Cindy Grimm for providing the Tree and Spiderweb models. They would also like to thank Cindy Grimm and Rachel Roberts for their helpful discussions and the anonymous reviewers for their constructive comments. This work was partly supported by the National Basic Research Program of China (Project 2006CB303106), the National Natural Science Foundation of China (Projects 60673004, 60333010, and 60225016), and the Basic Research Foundation of the Tsinghua National Laboratory for Information Science and Technology.

References

[1] H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Forms,* W. Wathen-Dunn, ed., pp. 362-380, MIT Press, 1967.

[2] N. Amenta, S. Choi, and R.K. Kolluri, "The Power Crust," *Proc. Sixth ACM Symp. Solid Modeling and Applications (SMA '01),* pp. 249-266, 2001.

[3] T.K. Dey and W. Zhao, "Approximate Medial Axis as a Voronoi Subcomplex," *Proc. Seventh ACM Symp. Solid Modeling and Applications (SMA '02),* pp. 356-366, 2002.

[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," *Proc. Int'l Conf. Computer Vision (ICCV '99),* pp. 828-834, 1999, 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 Analysis and Machine Intelligence,* vol. 14, no. 9, pp. 869-885, 1992.

[7] K. Palágyi and A. Kuba, "A Parallel 3D 12-Subiteration Thinning Algorithm," *Graphical Models Image Processing,* vol. 61, no. 4, pp. 199-221, 1999.

[8] G. Bertrand, "Simple Points, Topological Numbers and Geodesic Neighborhoods in Cubic Grids," *Pattern Recognition Letters,* 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," *J. Math. Imaging and Vision,* vol. 5, pp. 297-318, 1995.

[10] Z. Aktouf, G. Bertrand, and L. Perroton, "A 3D-Hole Closing Algorithm," *Proc. Sixth Int'l Workshop Discrete Geometry for Computer Imagery,* pp. 36-47, 1996.

[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, Aug. 2001.

[12] S. Bischoff and L. Kobbelt, "Isosurface Reconstruction with Topology Control," *Proc. Pacific Conf. Computer Graphics and Applications,* pp. 246-255, 2002.

[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. Medical Imaging,* vol. 20, no. 1, pp. 70-80, 2001.

[14] J. El-Sana and A. Varshney, "Controlled Simplification of Genus for Polygonal Models," *IEEE Visualization,* pp. 403-412, 1997.

[15] I. Guskov and Z.J. Wood, "Topological Noise Removal," *Graphics Interface,* pp. 19-26, 2001.

[16] M. Attene and B. Falcidieno, "Remesh: An Interactive Environment to Edit and Repair Triangle Meshes," *Proc. IEEE Int'l Conf. Shape Modeling and Applications (SMI '06),* p. 41, 2006.

[17] F.S. Nooruddin and G. Turk, "Simplification and Repair of Polygonal Models Using Volumetric Techniques," *IEEE Trans. Visual Computer Graphics,* vol. 9, no. 2, pp. 191-205, 2003.

[18] Z.J. Wood, H. Hoppe, M. Desbrun, and P. Schröder, "Removing Excess Topology from Isosurfaces," *ACM Trans. Graphics,* vol. 23, no. 2, pp. 190-208, 2004.

[19] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proc. 14th Ann. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '87),* pp. 163-169, 1987.

[20] D.W. Shattuck and R.M. Leahy, "Automated Graph Based Analysis and Correction of Cortical Volume Topology," *IEEE Trans. Medical Imaging,* vol. 20, no. 11, pp. 1167-1177, 2001.

[21] 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. Medical Imaging,* vol. 21, no. 2, pp. 109-121, 2002.

[22] A. Szymczak and J. Vanderhyde, "Extraction of Topologically Simple Isosurfaces from Volume Datasets," *IEEE Visualization,* pp. 67-74, 2003.

[23] S. Bischoff and L. Kobbelt, "Sub-Voxel Topology Control for Level-Set Surfaces," *Computer Graphics Forum,* vol. 22, no. 3, pp. 273-280, 2003.

[24] T. Ju, "Robust Repair of Polygonal Models," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 888-895, 2004.

[25] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual Contouring of Hermite Data," *Proc. 29th Ann. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '02),* pp. 339-346, 2002.

[26] G. Taubin, "A Signal Processing Approach to Fair Surface Design," *Proc. 22nd Ann. Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '95),* pp. 351-358, 1995.

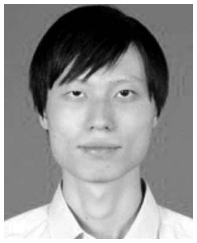
[27] J. Giesen and M. John, "The Flow Complex: A Data Structure for Geometric Modeling," *Proc. ACM-SIAM Symp. Discrete Algorithms (SODA '03),* pp. 285-294, 2003.

[28] A. Hatcher, *Algebraic Topology.* Cambridge Univ. Press, 2002.

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

[30] S. Bischoff and L. Kobbelt, "Structure Preserving CAD Model Repair," *Computer Graphics Forum,* vol. 24, no. 3, pp. 527-536, 2005.

[31] S. Bischoff and L. Kobbelt, "Topologically Correct Extraction of the Cortical Surface of a Brain Using Level-Set Methods," *Bildverarbeitung für die Medizin,* pp. 50-54, 2004.



Qian-Yi Zhou received the BS degree in computer science from Tsinghua University in 2005. He is currently an MS student in the Department of Computer Science and Technology at Tsinghua University. His research interests include computer graphics, geometry modeling, mesh processing, and computer-aided geometric design (CAGD).

Tao Ju received the BA degree in English and the BS degree in computer science from Tsinghua University in 2000 and the PhD degree in computer science from Rice University in 2005. He is currently an assistant professor in the Department of Computer Science and Engineering at Washington University in St. Louis. His research interests are in the areas of mesh processing, visualization, geometric modeling, and biomedical applications.

Shi-Min Hu received the PhD degree from Zhejiang University in 1996. He is currently a professor of computer science at Tsinghua University. His research interests include digital geometry processing, video-based rendering, rendering, computer animation, and computer-aided geometric design. He is on the editorial board of *Computer Aided Design*.