

Editing The Topology of 3D Models by Sketching

Tao Ju*
Washington University, St. Louis

Qian-Yi Zhou†
Tsinghua University, Beijing

Shi-Min Hu‡
Tsinghua University, Beijing

Abstract

We present a method for modifying the topology of a 3D model with user control. The heart of our method is a *guided topology editing* algorithm. Given a source model and a user-provided target shape, the algorithm modifies the source so that the resulting model is topologically consistent with the target. Our algorithm permits removing or adding various topological features (e.g., handles, cavities and islands) in a common framework and ensures that each topological change is made by minimal modification to the source model. To create the target shape, we have also designed a convenient 2D sketching interface for drawing 3D line skeletons. As demonstrated in a suite of examples, the use of sketching allows more accurate removal of topological artifacts than previous methods, and enables creative designs with specific topological goals.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations—Geometric algorithms, languages, and systems

Keywords: topology repair, sketching, skeleton

1 Introduction

The topology of a 3D solid model is characterized by its islands (i.e., connected pieces of solid), cavities (i.e., voids inside the solid) and handles (i.e., rings and tunnels). Ensuring that a model has the correct topology is crucial for a number of model-related tasks such as simplification, surface parameterization, and finite element analysis. Topology modification becomes necessary when artifacts, such as extraneous handles resulted from surface reconstruction, need to be removed (i.e., topology repair), or when the user desires a variation of the model with a different topology (i.e., shape design).

Prior to performing topology modification, we first need to know what *is* the objective topology. A commonly used objective is the *genus*, which is the number of handles minus the sum of the numbers of cavities and islands. However, having the right genus does not warrant a satisfactory result. In the example of Figure 1, the original model (a) contains an erroneous handle (hence is genus-1), which is removed in both (b,d). However, (b) is unsatisfactory as the paper clip is cut open in the middle. The problem is worsened when both the source model and the objective topology contain handles, cavities or islands, where the desired set of topological features would be difficult to be characterized precisely and robustly using genus or using any numerical measures of feature sizes (see more such examples in Section 5).

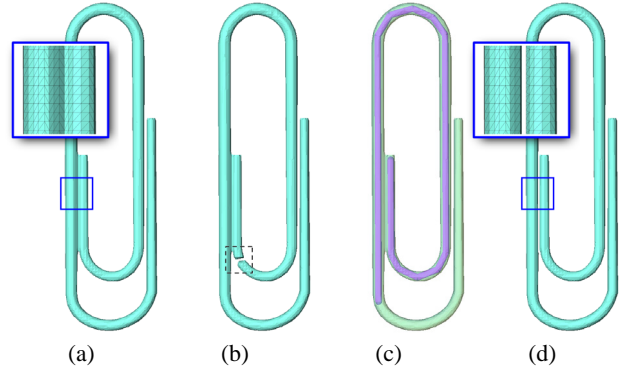


Figure 1: The Clip model (a) incorrectly merges two parts (highlighted) and has genus 1. A typical topology repair method would produce a model with the right genus (b) yet cutting the model at an undesired location. Our approach utilizes a simple target (purple in c) created by sketching and produces a satisfactory result (d).

In this work we adopt a target shape provided by the user as our objective in topology modification. Our motivation is that a large number of 3D models used in practice have a simple topology known to the user, and that the user is the ultimate judge of correctness. While the target can assume any geometric shape, we have designed a convenient 2D sketching interface where the user may easily create 3D line skeletons without having to tediously specify the 3D locations of end-points. An example of the sketched lines is shown in Figure 1 (c).

Given a source and a target, the core of our method is a *guided topology editing* algorithm that produces a modified source model topologically consistent (to be defined in Section 3) with the target. Guided by the target, our algorithm applies topological changes in a sequential manner, and each change is made with minimal modification to the source model. In the example of Figure 1, the use of a target in (c) combined with the guided topology editing algorithm allows accurate removal of the artifact handle, shown in (d).

Contributions: While sketching has been widely used in editing the geometry of 3D models, we introduce a similar user guidance into topology editing and develop a sound mathematical foundation for topology editing guided by a user-defined target. This foundation is built on top of a previous method [Zhou et al. 2007], which utilizes the skeleton of a solid model for removing topological handles without user intervention. Comparing to [Zhou et al. 2007] and previous approaches, we present three key theoretical contributions:

1. While existing topology repair approaches are typically restricted to removing topological handles, we show that a full range of topological modifications, including addition and removal of islands, cavities and handles, can be easily achieved by modifying a skeleton representation of the model.
2. Given any user-specified target, our method automatically modifies the source 3D model to be topologically consistent with the target. To the best of our knowledge, this is the first method with such topological guarantee, which is essential for the purpose of user-guidance.
3. While most volume-based techniques including [Zhou et al. 2007] modifies a source model solely by expanding or by shrinking, our method considers both expanding and shrink-

*e-mail: taoju@cs.wustl.edu

†e-mail: zqy@mails.tsinghua.edu.cn

‡e-mail: shimin@tsinghua.edu.cn

ing when making each topology change, hence preventing unnecessarily large volume modifications.

2 Previous work

Topology repair and simplification With the exception of a few methods that globally alter the shape of the input model, such as the multi-resolution simplification approach of [Andujar et al. 2002] and the morphological approach of [Nooruddin and Turk 2003], the majority of topology repair methods are designed to simplify the topology of the input model by locally removing redundant handles. Common to these methods is the use of either genus or some measure of feature size in describing the objective topology. For example, [El-Sana and Varshney 1997] identifies small tunnels as regions not accessible to a sphere of user-specified radius rolling on the surface. The method of [Guskov and Wood 2001] removes small handles that are completely contained in a mesh neighborhood of a given size. Using a volumetric representation, [Szymczak and Vanderhyde 2003] applies topology-preserving carving operations to extract iso-surfaces with the desired genus. [Wood et al. 2004] detects each surface handle as a cycle in the Reeb graph of an iso-surface, whose size is measured as the shortest geodesic circle around the handle. Similar graph approaches were adopted in the work of [Shattuck and Leahy 2001], [Han et al. 2002] and [Zhou et al. 2007], where a handle corresponds to a cycle in the graph encoding the topology of the solid (instead of the surface), and the size of the handle is measured as the thickness of the solid.

Another common difficulty for most existing approaches is to ensure the geometric or topological correctness of their results. For example, mesh-based approaches such as [Guskov and Wood 2001] may introduce geometric errors in the form of self-intersecting polygons, while volume-based approaches may introduce new handles as the result of removing existing ones [Nooruddin and Turk 2003; Wood et al. 2004] or may not be able to identify and remove all handles for models with complex topology [Zhou et al. 2007].

Surface reconstruction with topology control This class of methods [Mangin et al. 1995; Aktouf et al. 1996; Kriegeskorte and Goebel 2001; Bischoff and Kobbelt 2002; Sharf et al. 2006] is designed to reconstruct iso-surfaces from volumetric data with a known topology type. Starting from an initial solid with the correct surface topology, these methods grow the solid so that the grown solid is topologically consistent with the initial one. While such methods have been effective in reconstructing genus-0 models, application to non-spherical cases has been limited due to the difficulty in defining the initial solid. While the method of [Sharf et al. 2006] allows topology change in the growing process by merging growing fronts, this method suffers from the same drawback as topology simplification methods as the merging is controlled by a size parameter. Furthermore, surface reconstruction by growing can only result in a subset of the original model. Hence these methods may introduce a much larger modification to the original model than necessary, for example, when thin tunnels or large islands are present and a genus-0 topology is desired.

Sketching interfaces Sketching has emerged as a popular interaction technique in recent years. A number of sketching tools, such as the Teddy system presented by [Igarashi et al. 1999], allow creation and editing of 3D models using 2D line sketches. In animation, [Davis et al. 2003] use 2D sketches as queries to find matching 3D poses in order to speed up animation creation. Sketches have also been used to guide geometric editing and deformation of meshes by [Nealen et al. 2005] and [Kho and Garland 2005]. In these methods, the sketch interface back-projects 2D screen lines onto the surface of a 3D model or a tangent plane on the surface by computing camera projection of mesh elements or performing ray intersection with the surface. In our work, the 2D lines are used

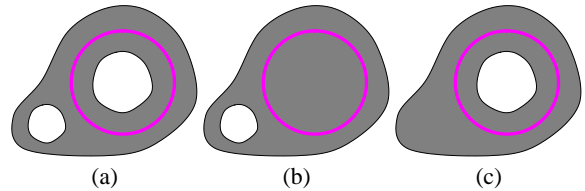


Figure 2: Topological consistency between the source (gray) and the target (purple): while both modifications (b,c) of the source (a) have a single handle, only (c) is topologically consistent with the target. (See definition in Section 3.)

to create 3D skeletons lying *interior* to a 3D model, which is then utilized for topology editing.

3 Guided topology editing

We first present a general algorithm for modifying a source model guided by some subset of the source, called the target. We call this process *guided topology editing*. Note that the target model can be any subset of the source and is not restricted to the line shapes that the user will be able to draw using our interface (discussed in the next section). Our goal is to produce a modified source model that is topologically consistent with the target. Here, consistency implies that the modified source not only has to have the same genus as the target, but can also be continuously retracted onto the target without changing its genus in the retraction process¹. As illustrated in Figure 2, the retractability requirement is necessary to differentiate two shapes with the same genus.

As shown in Figure 3, our algorithm proceeds in two stages. Given a source model represented as a discrete volume and a target model as a subset of the source (Figure 3 (a)), our algorithm first computes a subset of the source, known as a *fat skeleton* (or FS), which contains the target while being topologically consistent with the source (Figure 3 (b)). Next, the FS is reduced to the target through a sequence of reduction operations. Each reduction removes a portion from the FS and modifies the source accordingly, if necessary, to maintain its topological consistency with the reduced FS (two of these operations are shown in Figure 3 (c,d)). Upon completion, the modified source model is topologically consistent with the remainder of the FS, which is the target (Figure 3 (e)).

Our algorithm guarantees topological consistency given any target, which may have a different number of handles, cavities or islands as the source. In addition, the algorithm identifies the least modification to the source, either additions (Figure 3 (c)) or subtractions (Figure 3 (d)), that is necessary to maintain topological consistency after each FS reduction. This method builds upon the skeleton-based handle-removal approach in [Zhou et al. 2007], while substantially extending the theoretical results of the latter to form a complete framework for guided topology editing. Specifically, our method allows a full range of topological editions through skeleton modification (Section 3.3.1), considers both addition and subtraction when making each topological change (Section 3.3.2), and guarantees to meet the user’s demand using an novel iterative routine (Section 3.3.3). These extensions are detailed in Section 3.3. For completeness, we will start by briefly describing the data representation (Section 3.1) and FS generation (Section 3.2), which are similar to those in [Zhou et al. 2007].

3.1 Cell complexes

To facilitate computing and manipulating fat skeletons, we represent a solid model as a collection of points, edges, faces and cells

¹In algebraic topology, this retraction is known as *deformation retraction* [Hatcher 2002], a continuous map $f : X \times [0, 1] \rightarrow Y$ where $Y \subset X$ such that $f(x, 0) = x$ and $f(x, 1) \in Y$ for $x \in X$, and $f(x, t) = x$ for $x \in Y$ and $t \in [0, 1]$.

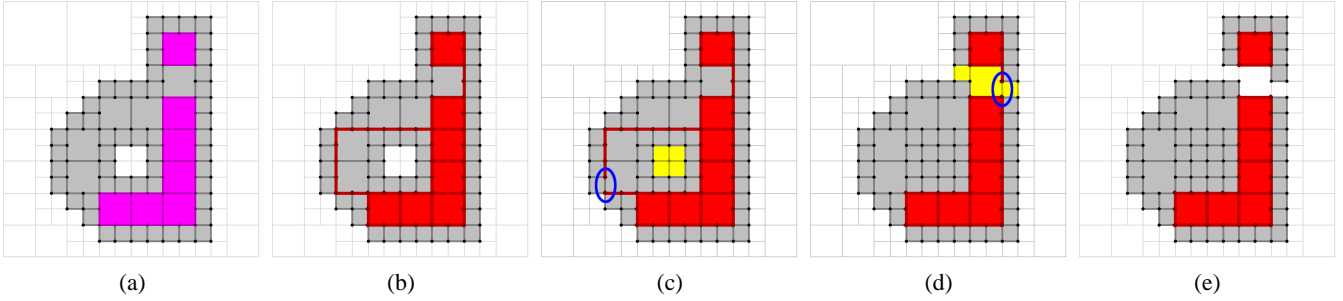


Figure 3: Guided topology editing. (a): The source (gray) and target (purple) represented as cell complexes. (b): An FS (red) of the source that contains the target. (c,d): FS reductions that modify the source to maintain topological consistency with the reduced FSs (modifications shown in yellow). (e): When the FS is reduced to the target, the modified source is topologically consistent with the target.

on a 3D grid. In particular, we require that if an element f belongs to this collection and f contains some element e (for example, each face on a uniform grid contains four grid edges and four grid points), then e also belongs to the collection. Such collection of grid elements is called a *cell complex*. In our work, both the source and the target models are represented as cell complexes on an adaptive octree grid, as illustrated in 2D in Figure 3. The main advantages of using the cell complex representation, to be revealed in more details soon, are a simple procedure for generating topologically consistent subsets (Section 3.2) and the flexibility in representing isolated grid elements that make various topological modifications possible (Section 3.3).

The conversion between a cell complex and other mainstream solid representations, in particular, a polygonal mesh, was detailed in [Zhou et al. 2007]. In short, to construct a cell complex, the mesh is first scan-converted into an adaptive inside/outside volume, and the cell complex is the collection of all inside grid points and all grid elements that contain only inside points. Note there are a number of robust scan-conversion algorithms [Ju 2004; Bischoff et al. 2005] capable of handling arbitrary meshes (or triangle soups). To preserve the geometric features of the polygonal mesh, sampled mesh points are stored on the cell complex, which can be used to reconstruct the mesh from the modified cell complex using an adaptive iso-surfacing routine [Zhou et al. 2007].

3.2 FS generation

One advantage of representing a solid model as a cell complex is that it permits a simple procedure for generating topologically consistent subsets. Let D give the dimensionality of an element such that $D(e) = 0, 1, 2, 3$ when e is a point, edge, face or cell. A *simple removal* from a cell complex V is the deletion of a pair of elements $\{\delta, \sigma\} \subset V$ such that $D(\delta) = D(\sigma) + 1$ and that δ is the only element in V that contains σ . For example, σ can be a grid edge that is contained in no other elements of V except a grid face δ . We define a *fat skeleton* (FS) of V as the remainder of V after a sequence of simple removals. An example of a sequence of simple removals and the FSs after each removal are shown in Figure 4.

It is not hard to show that an FS is also a cell complex [Zhou et al. 2007]. More importantly, a cell complex V is topologically consistent with an FS of V , since each simple removal simulates a continuous retraction² that does not alter the genus of V .

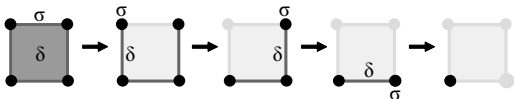


Figure 4: A sequence of simple removals from a cell complex.

²Formally, a simple removal is a type of *elementary collapse*, which is a discrete example of deformation retraction [Matveev 2003]

In the first stage of our algorithm, we compute an FS of the source model, which contains the target model. To reduce the computation later in our algorithm, we wish to minimize the size of the resulting FS. To this end, we adopt the iterative thinning procedure in [Zhou et al. 2007], which performs simple removals to a cell complex on an adaptive grid in a layer-by-layer manner until no more simple removals are possible. We modify this procedure by protecting elements of the target model from being removed during thinning.

3.3 FS reduction

In the second stage, we will modify the source model to be topologically consistent with the target. This is achieved by FS reductions. Given a triple of cell complexes $\{V, S, K\}$ where $K \subset S$ and S is a skeleton of V , an FS reduction produces a new triple $\{V', S', K\}$ where $K \subseteq S' \subset S$ and S' is an FS of V' . Starting with V as the source model, S as the initial FS computed in the first stage, and K as the target model, a finite sequence of FS reductions will reduce the initial skeleton to the target while modifying the source to adopt the target as its FS (hence topologically consistent to the target).

A trivial form of FS reduction is a simple removal, where S' is still an FS of the original cell complex V (i.e., $V' = V$). More generally, we consider FS reduction by removing a single element $e \in S$. Note that e is necessarily *isolated*, meaning that no other elements in S contains e , so that the remainder $S' = S \setminus \{e\}$ is still a cell complex. In this general case, S' is no longer the FS of V . In the following, we demonstrate two possible ways of computing the modified cell complex V' by either subtracting from or adding to V so that S' is an FS of V' .

3.3.1 Subtraction

We first consider shrinking a cell complex as a result of removing an isolated element from its FS. Let $R_{V,S}$ be the sequence of simple removals that reduce V to S , which consists of pairs $\{\delta, \sigma\}$ where $D(\delta) = D(\sigma) + 1$ and δ contains σ . We define:

Definition 1 A path associated with an isolated element $e \in S$ is any sequence $\{\sigma_0, \delta_1, \sigma_1, \dots, \delta_k, \sigma_k\} \subseteq V$ where $k \geq 0$ such that $\sigma_0 = e$ and, for all $1 \leq i \leq k$, $\{\delta_i, \sigma_i\} \in R_{V,S}$ and δ_i contains σ_{i-1} .

Definition 2 The generating set of an isolated element $e \in S$, denoted as $W_{V,S}(e)$, is the union of all paths associated with e .

Intuitively, the generating set consists of the subset of simple removals reducing V to S that contribute to the “isolation” of e , or the removal of elements in S that contain e . We illustrate the generating set of an isolated edge in 2D in Figure 5 (a). We next show:

Proposition 1 Let S be an FS of cell complex V and e be an isolated element of S . Then $V' = V \setminus W_{V,S}(e)$ is a cell complex, of which $S' = S \setminus \{e\}$ is an FS.

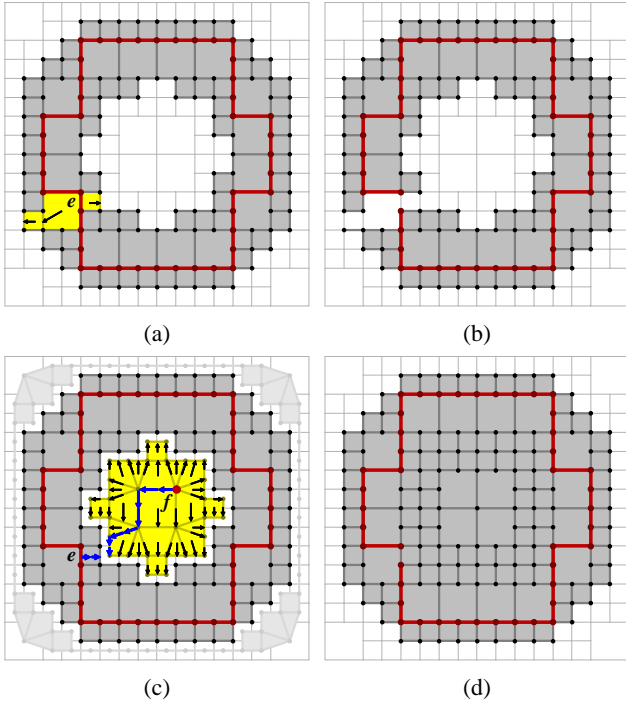


Figure 5: FS reduction. (a): An isolated edge e on the FS (red) and its generating set (yellow and showing the simple removals $\{\delta, \sigma\}$ in the set as arrows $\delta \rightarrow \sigma$). (b): Subtracting the generating set of e from the ring. (c): A compatible isolated point f on the FS of the complement (both inside and outside the ring), its generating set, and the unique paths associated with e and f (blue). (d): Adding the dual of the generating set of f to the ring.

Proof: Observe that if $f \in V$ contains some element in a path associated with e , f is also in a path associated with e . Hence $f \in V'$ only if all elements f contains are in V' , implying V' is a cell complex. S' is an FS of V' , as V' can be reduced to S' using the simple removals in $R_{V,S}$ that are not in $W_{V,S}(e)$. \square

In Figure 5 (b), the subtraction of the generating set breaks the handle ring. Note that the generating set was first introduced in [Zhou et al. 2007], where it was defined differently and restricted to isolated edges. Our extension allows e to assume different type of isolated elements. In particular, removing e from S (and subtracting $W_{V,S}(e)$ from V) results in a variety of topological changes, including losing an island (when e is a point), gaining an island or losing a handle (when e is an edge), gaining a handle or losing a cavity (when e is a face), and gaining a cavity (when e is a cell).

3.3.2 Addition

Besides shrinking, it is often possible to expand a cell complex V to admit the remainder $S' = S \setminus \{e\}$ as its FS. For example, the handle in Figure 5 (a) can be removed either by breaking the handle ring (Figure 5 (b)) or by filling the interior hole (Figure 5 (d)). To achieve such expansion, we consider the complement of the cell complex V and its FS. This complement, \bar{V} , is another cell complex defined as $\widehat{G \setminus V}$ [Zhou et al. 2007]. Here, G is the set of all grid elements (including an infinite outside cell), and $\widehat{\cdot}$ is a dual operator such that \widehat{e} is formed by connecting the centroids of the grid cells containing or equal to e , and that $\widehat{\widehat{e}} = e$. In 3D, the dual of a grid point, edge, face and cell respectively have dimensions 3, 2, 1 and 0. A 2D example of this complement is shown in Figure 5 (c). Let T be the FS of \bar{V} , we consider:

Definition 3 Isolated elements $e \in S$ and $f \in T$ are compatible if

$D(\widehat{f}) = D(e) + 1$ and there are unique paths $\{e, \dots, \sigma_e\} \subseteq V$ and $\{f, \dots, \sigma_f\} \subseteq \bar{V}$ associated with e and f such that $\widehat{\sigma_f}$ contains σ_e .

To illustrate the definition in 2D, we highlight two compatible elements and their paths in Figure 5 (c), where e is an isolated grid edge and f is an isolated point (whose dual \widehat{f} in 2D is a grid face). Note that there is only one path between e and f that satisfies the properties in Definition 3. We further observe that either subtracting the generating set $W_{V,S}(e)$ from V (Figure 5 (b)) or adding the dual of the generating set $W_{\bar{V},T}(f)$ to V (Figure 5 (d)) will remove the handle and adopt the remainder $S' = S \setminus \{e\}$ as the FS of V . More generally, we prove (in Appendix A):

Proposition 2 Let S, T be the FSs of cell complexes V, \bar{V} , and $e \in S, f \in T$ be two compatible isolated elements. Then $V' = V \cup \widehat{W_{\bar{V},T}(f)}$ is a cell complex, of which $S' = S \setminus \{e\}$ is an FS.

In 3D, the isolated element e may assume a grid point, edge or face, whose compatible element f by definition would be an isolated face, edge or point on the complement FS T . Subtracting $W_{V,S}(e)$ from V or adding the dual of $W_{\bar{V},T}(f)$ to V would achieve a same topological change by different means, such as losing an island by either removing the island or bridge two islands (when e is a point), losing a handle by either breaking the handle ring or filling the handle hole (when e is an edge), and losing a cavity by either bridging two cavities or filling the cavity (when e is a face). Note that, however, not every isolated element on the FS may have a compatible element on the complement FS. For example, creating a new tunnel by subtracting $W_{V,S}(e)$ from V when e is a face may not be achieved by adding the dual of $W_{\bar{V},T}(f)$ to V for any element f on the complement FS T .

3.3.3 Iterative FS reductions

To reduce the initial FS to the target, FS reductions are invoked iteratively. To minimize the amount of modification to the source, we select the FS reduction that would introduce least modification at each iteration. Given a triple $\{V, S, K\}$, this is performed by computing a weight $w(e)$ for each isolated element $e \in S \setminus K$ as the minimal volume that would be subtracted from or added to V if e is removed from S . The element e with the minimum weight is removed (if no more simple removals can be made), and V is updated by either subtracting or adding, whichever modifies V least. The algorithm terminates when $S = K$, when the modified V becomes topologically consistent with the target.

Based on the above discussions, the weight $w(e)$ is computed as the minimum of $|W_{V,S}(e)|$ and $|\widehat{W_{\bar{V},T}(f)}|$ for all compatible elements f of e (if f exists), where $|\cdot|$ is the total volume occupied by a set of grid elements. We consider every non-cell grid element occupying an ε -thick space disjoint from the space of other elements. Hence the volume of the space is ε^3 for a point, $\varepsilon^2(l - \varepsilon)$ for an edge of length l , $\varepsilon(l - \varepsilon)(w - \varepsilon)$ for a face of size $l \times w$, and $(l - \varepsilon)(w - \varepsilon)(h - \varepsilon)$ for a cell of size $l \times w \times h$. In our implementation, $\varepsilon = 0.5$ is used (assuming leaf cells of the octree have unit size).

The pseudo-code of the iterative routine is presented in Figure 6. Note that the routine utilizes the FS of the complement \bar{V} , T . To avoid computing T from \bar{V} every time V is updated, we only compute T once when the initial FS S is generated. During FS reduction, T is updated to remain as an FS of \bar{V} . Observe from Propositions 1 and 2 that if T contains some element f compatible with the isolated element e removed from S , it suffices to update T by simply subtracting f . If no compatible elements exist, we note that V is an FS of \bar{T} (see Lemma 1 in Appendix A), hence S is also an FS of \bar{T} . Adding the dual of the generating set $W_{\bar{T},S}(e)$ to T yields an FS of the complement $\bar{V} \setminus \widehat{W_{V,S}(e)}$ (see Lemma 2 in Appendix

```

// Source  $V$ , FS  $S, T$ , Target  $K \subseteq S$ 
Repeat until  $S = K$ 
  If  $\exists$  a simple removal  $\{\delta, \sigma\} \subset S \setminus K$ 
     $S \leftarrow S \setminus \{\delta, \sigma\}$ 
  Else If  $\exists$  a simple removal  $\{\delta, \sigma\} \subset T$ 
     $T \leftarrow T \setminus \{\delta, \sigma\}$ 
  Else
     $e \leftarrow$  IE in  $S \setminus K$  with minimum  $w(e)$ 
     $S \leftarrow S \setminus \{e\}$ 
     $f \leftarrow$  Compatible IE in  $T$  with minimum  $|W_{\bar{V}, T}(f)|$ 
    If  $f = \emptyset$ 
       $V \leftarrow V \setminus W_{V, S}(e)$ 
       $T \leftarrow T \cup \widehat{W_{\bar{V}, S}(e)}$ 
    Else
      If  $|W_{V, S}(e)| < |W_{\bar{V}, T}(f)|$ 
         $V \leftarrow V \setminus W_{V, S}(e)$ 
      Else
         $V \leftarrow V \cup \widehat{W_{\bar{V}, T}(f)}$ 
       $T \leftarrow T \setminus \{f\}$ 

```

Figure 6: Pseudo code for iterative FS reductions (IE stands for isolated element).

A). Finally, to maintain the minimality of T , simple removals are performed whenever possible.

To efficiently compute the paths (and hence generating sets and compatible elements) on an adaptive octree, we maintain pointers between elements in each simple removal when S and T are first computed. During FS reduction, the paths can be found in a breadth-first search starting from an isolated element using fast recursive neighbor-finding procedures [Ju et al. 2002]. Furthermore, to avoid re-computing the weights of each isolated element after each iteration, the weights only have to be updated locally, after removing an isolated element e , for elements of S whose generating sets intersect that of e and for elements of T compatible with e .

4 A sketching interface

Editing the topology of a 3D model needs a target shape. Ideally, the user should be able to create the target quickly from the source model. As a first step towards this goal, we have designed a unique interface for drawing skeleton-like targets. While such skeletons can be created in animation design tools, users are typically required to provide the 3D location of each skeleton joint, which can be tedious. In contrast, our interface lets the user sketch 2D lines on the computer screen while automatically “back-projecting” the sketch to the interior of the 3D model at interactive rates.

4.1 User interaction

We first show how a user can quickly draw 3D lines in our 2D sketching interface using the example of Figure 7. Given a 3D model, the user may choose any view of the model to start sketching. The model is then rendered semi-transparently to ease viewing, shown in Figure 7 (a). As the user moves the mouse over the screen, the layers of solid materials projected to the mouse location are displayed in realtime (besides the cursor). By scrolling the mouse wheel, a desired layer can be selected easily without interrupting sketching. When combined with a hot key, the user may also adjust a depth value within each layer (indicated by the blue arrow beside the layers), initially set to the middle of that layer. Depth adjustment can be useful when drawing overlaying sketches within a same layer of solid. Each click of the left mouse button places a new point at the selected depth in the selected layer of solid and forms a line from the previously placed point, and clicking the right mouse button stops line connection. In addition, new points can be snapped onto nearby, existing points and lines, making it easy for

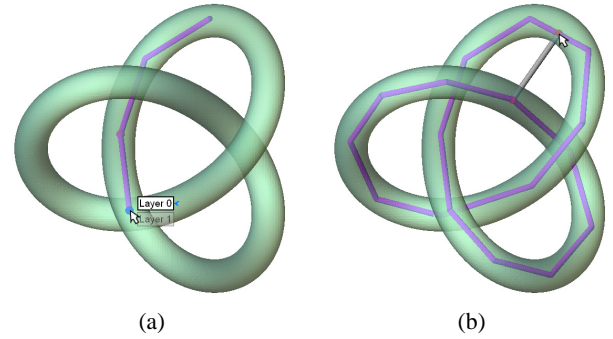


Figure 7: The sketching interface. See description in Section 4.1.

creating junctions and closures, as shown in (b). Although seldom necessary in practice, the user may continue sketching on different views of the model.

4.2 Depth peeling

The key that enables fast drawing of 3D lines on a 2D screen is the knowledge of depths of different layers of solids projected on to the mouse location. For efficiency, such depth information is obtained by using *depth peeling*. Depth peeling is a fragment-level depth sorting technique [Mammen 1989; Diefenbach 1996] originally developed to avoid the need to pre-sort polygons in rendering transparent models by rendering the same scene in multiple passes. In particular, the k th pass renders the k th nearest fragment at each pixel, and the depth of that fragment is kept in the depth buffer. In our interface, values in the depth buffer after each pass are extracted into a depth array at each pixel. For a closed model, each pair of values in the array represents the near and far depth of a layer of solid projected onto that pixel. Since depth peeling is performed only once after a view is chosen, this approach allows real-time sketching interaction regardless of the size of the input model.

4.3 Conversion to cell complexes

To use with our topology editing algorithm, the sketched line skeleton is then converted to the cell complex representation. Given the source model represented as a cell complex on an octree grid, the target cell complex consists of finest-level grid cells in the octree that intersect with the skeleton lines, as well as all grid elements contained by those cells. Note that the original octree cells that are not at the finest level and intersect the lines are subdivided in this process. Observe from Figure 7 (b) that some parts of the line skeleton (colored gray) created by our interface may lie outside the source model. This is useful, for example, when the user wishes to bridge between islands or create additional handles. To handle target cell complexes that are not subsets of the source cell complex, we simply treat the union of the target and source as the new source cell complex, after which the algorithm presented in Section 3 can be directly applied.

5 Results

In this section, we collect a suite of examples to illustrate the guided topology editing algorithm and the usefulness of sketching. For each example, a polygonal mesh is imported to our sketching interface, where 3D lines are drawn as the target. Both the mesh and the lines are converted to cell complexes, and the resulting mesh is reconstructed from the modified cell complex. The conversion procedures are discussed in Sections 3.1 and 4.3.

We first use a simple example to illustrate guided topology editing in 3D. Given the Clip model in Figure 1, the two stages (thinning and FS reduction) of our algorithm are shown respectively in (a,b) and in (c,d) of Figure 8. In particular, the isolated edges in the initial FS (see insert of (b)) are removed during FS reduction, resulting the

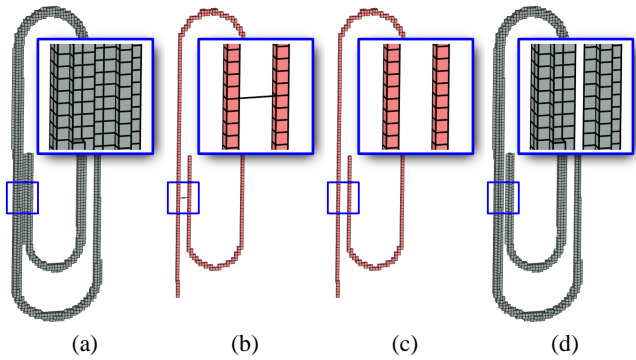


Figure 8: Guided topology editing of the Clip model in Figure 1: the source cell complex (a), the initial FS (b), the reduced FS (i.e., the target cell complex) (c), and the modified cell complex (d).

separation of the two tubes of the clip (see insert of (d)).

We next use a number of CAD and scanned models (Figures 10 to 13) to demonstrate the use of sketching in topology repair and design. In particular, we hope to elucidate the advantage of our method over previous topology repair and topology-controlled surface reconstruction approaches.

A major benefit of using sketching in topology repair is that it lets the *user* select the topological features to be preserved. As shown in Figure 10, sketching is more direct and precise in describing the objective topology than using either genus or measures of feature size. The importance of user-selection is amplified in Figure 11, where “thinner” handles (e.g., the arms of the mother), which would have been removed in most topology repair methods, are selected by the user to be preserved. Using sketching, our method guarantees that the edited model not only has the correct number of topology features but also has the set of features that the user desires.

While most topology repair methods are designed for removing handles, our method can be used to perform additional topology operations that include (but not limited to) creating new handles (Figure 12), reducing islands through bridging (Figure 9), and creating islands by separating wrongly connected pieces (Figure 13). In particular, the separation of the 24 bones in the Foot model in Figure 13 suggests a possible application of our method in user-guided segmentation of 3D models. The variety of topological changes offered by our method can be particularly useful in shape design with a specific topological goal, as illustrated in both Figures 12 and 9.

We note that the user-provided targets in our method can also be used with previous topology-controlled surface reconstruction methods, which would grow the target towards the model surface in a topology preserving manner. However, since these methods modify the model by subtraction, they may introduce a much larger modification than necessary in the presence of large islands (e.g., the island in letter “e” in Figure 9) or small tunnels (e.g., the smaller hole in Figure 10). On the other hand, our guided topology algorithm selectively applies addition or subtraction for least modification. In particular, large islands are bridged instead of removed (Figure 9), and handles with small holes are filled instead of cut on the side (Figure 10 (c,d), comparing with (a,b)).

Finally, we demonstrate the efficiency of our method on large scanned models in Figure 14. Observe that, even at very high grid resolution (2^{12}), the entire topology editing process completes in an hour. In addition, the creation of the sketches in all examples in this paper never took more than a few minutes using our interface. The performance of our implementation on all examples in the paper are summarized in Table 1. All tests were run on a consumer-level PC with P4 3Ghz CPU and 2G memory.



Figure 9: An example of topology design: the source model (a) (a plate with words “San Diego” carved out) is different from the target (purple in (b)) in both the number of handles and islands. After editing, both the plate and the words become a single connected component (c). Editing involves both subtraction (for breaking handles) and addition (for bridging islands), highlighted in (c), to guarantee a topologically consistent result with minimal change.

6 Conclusion and discussion

We have described a user-guided method for editing the topology of a 3D model. The system consists of two components, a convenient sketching interface for drawing 3D line shapes, and an efficient and robust algorithm for modifying a 3D model to be topologically consistent with a target shape. Our algorithm allows a variety of topological changes besides handle removal, and each change is made with minimal modification to the original model.

Our current algorithm and interface have several limitations that we wish to address in our future work:

Sketching complex topology While the guided topology editing algorithm is capable of handling any user-specified targets, the current line-sketching interface is too simple for specifying complex targets. One of our primary future tasks is to expand the interface to fully exploit the power of the underlying algorithm. First, our interface currently does not support drawing of closed surfaces, hence cannot be used to create new cavities (or preserve existing cavities). Besides adding surface-sketching functionality, we will consider more convenient ways for specifying cavities, such as sketching the topology of the complement space. Second, sketching the topology of a complex model (e.g., the spokes of a wheel) can be time-consuming. To simplify the task, we will look into incorporating advanced editing tools (e.g., copying and pasting) and automatic completion algorithms (e.g., extension of example-based geometry completion). We can also visualize the skeleton of the source model to guide sketching or to create a complex sketch by locally modifying the skeleton. Third, while the depth peeling approach works well for a small number of surface layers (which is the case for all our test models), we will experiment with extremely noisy models and device better approaches if necessary to minimize cluttering on

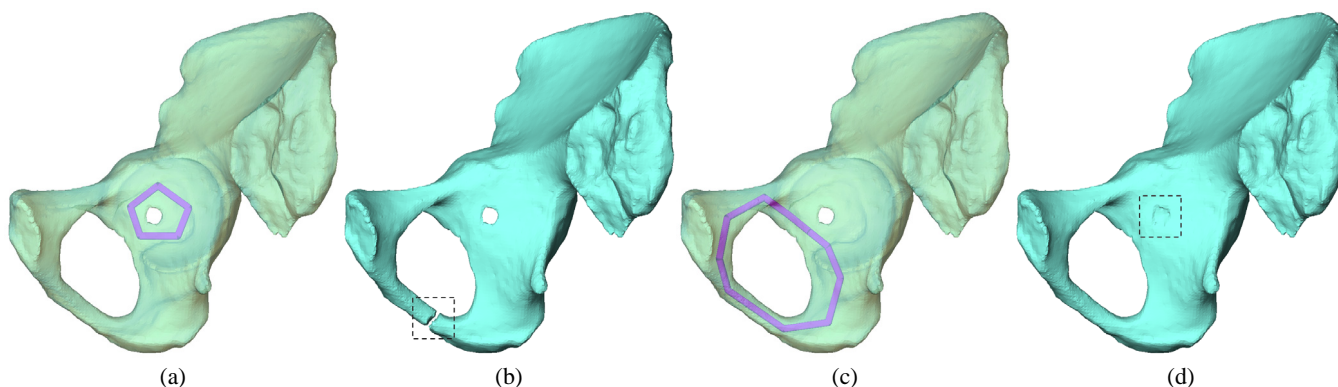


Figure 10: Editing the topology of the Hip model with different targets (a,c) results in the removal of different handles. Each handle is removed by breaking the handle ring (b) or filling the handle hole (d), whichever involves less modification to the model.

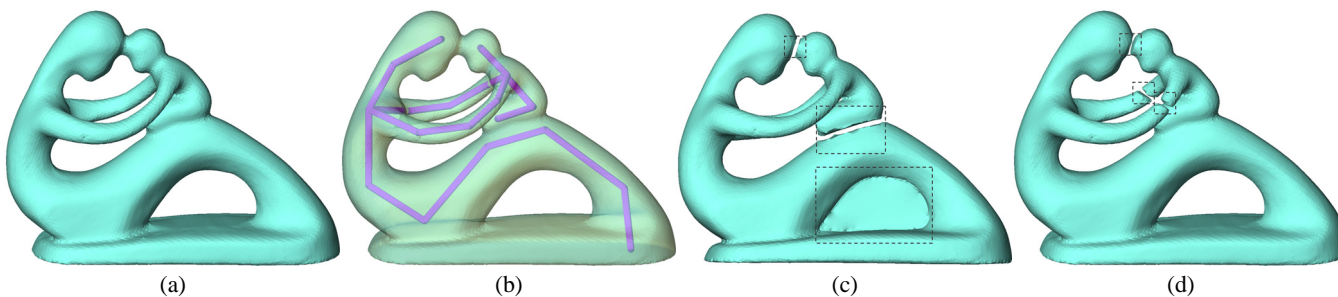


Figure 11: Editing the topology of the Mother model (a) with the target (b) selectively removes handles that user desires. Without the target, a typical topology repair method would identify thin handles to be removed (d), yielding an undesirable result.

the screen. Last but not least, we are planning to design user-tests to see how well novice users understand and perform the task.

Shape control in topology editing In our current algorithm, the resulting shape change is automatically determined with the goal of minimizing the amount of volume added or removed. However, it may be desirable to have the user control the shape change. For example, the user may desire to use addition in certain places even if subtraction would induce smaller volume change, to create a new tunnel with a certain width (instead of a thin hole in Figure 12 (b)), or to replace small handles by a smooth surface. Such shape control can be made possible by associating sketches with shape properties. For example, a colored sketch can be used to indicate desired solid thickness or surface smoothness. The current guided topology editing algorithm will be modified to minimize a weighted sum of the modified volume and deviation from the sketch shape.

Geometric fidelity To process triangular meshes, we currently require the meshes to be converted to and from a volumetric grid, which may result in loss of geometric details. Using feature-sensitive scan-conversion [Ju 2004] and iso-surfacing techniques [Ju et al. 2002], we are able to retain features (e.g., sharp edges and corners) on the original mesh in an accurate manner, as shown in the example of Figure 12. Note that our method can be easily extended to further preserve the tessellation on the original mesh. In particular, we note that our guided topology editing algorithm works on cell complexes defined in any structured or unstructured 3D grids. As a result, application onto a tetrahedral grid that contains both the triangles of the original mesh and the lines in the target shape would retain the original triangles as part of the resulting edited mesh.

7 Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments. The models used in this paper are courtesy of the Stanford 3D Scanning Repository (Figure 14 (a,c,d,e)), Aim@Shape Digital Shape WorkBench (Figures 11, 13, 14 (b)), Cyberware (Fig-

ure 10) and 3DM3.com (Figure 1). This work was supported in part by the National Basic Research Project of China (Project Number 2006CB303106) and the Natural Science Foundation of China (Project Number 60673004, 60333010).

References

- AKTOUF, Z., BERTRAND, G., AND PERROTON, L. 1996. A 3d-hole closing algorithm. In *6th International Workshop on Discrete Geometry for Computer Imagery*, 36–47.
- ANDUJAR, C., BRUNET, P., AND AYALA, D. 2002. Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. Graph.* 21, 2, 88–105.
- BISCHOFF, S., AND KOBBELT, L. 2002. Isosurface reconstruction with topology control. In *Pacific Conference on Computer Graphics and Applications*, 246–255.
- BISCHOFF, S., PAVIC, D., AND KOBBELT, L. 2005. Automatic restoration of polygon models. *ACM Trans. Graph.* 24, 4, 1332–1352.
- DAVIS, J., AGRAWALA, M., CHUANG, E., POPOVIĆ, Z., AND SALESIN, D. 2003. A sketching interface for articulated figure animation. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 320–328.
- DIEFENBACH, P. J. 1996. *Pipeline rendering: interaction and realism through hardware-based multi-pass rendering*. PhD thesis, Philadelphia, PA, USA.
- EL-SANA, J., AND VARSHNEY, A. 1997. Controlled simplification of genus for polygonal models. In *IEEE Visualization*, 403–412.
- GUSKOV, I., AND WOOD, Z. J. 2001. Topological noise removal. In *Graphics Interface*, 19–26.

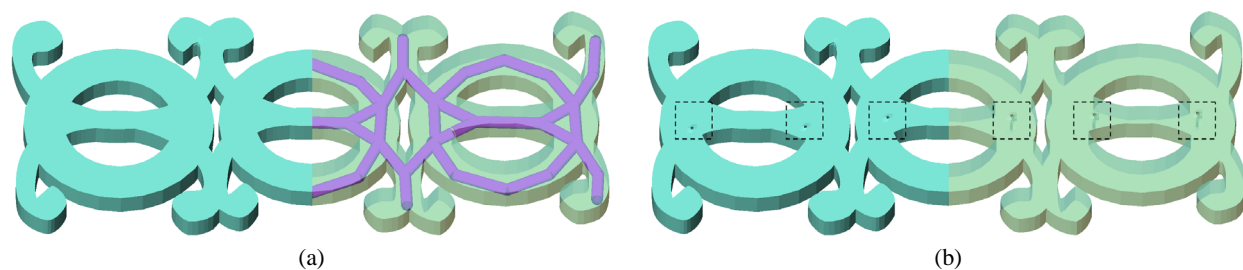


Figure 12: Sketch can be used not only to remove handles, but also to introduce new handles: the sketch drawn in (a) (only showing the symmetric half) introduces six handle holes to the edited model (b) (transparency is drawn to show holes penetrating the plate). Note in addition that geometric features (e.g., sharp edges and corners in (a)) are well preserved after editing.

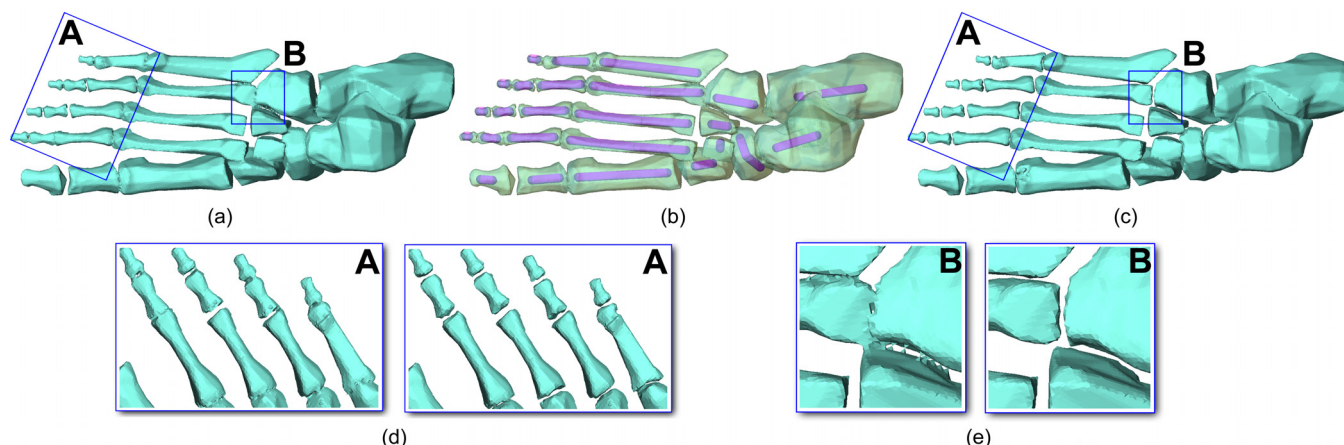


Figure 13: Besides handles, sketching can be used for making other topological changes: the foot model (a) incorrectly merges neighboring bones, which are subsequently separated into 24 pieces (c) using the target in (b). Close-up looks are shown in (d,e)

HAN, X., XU, C., BRAGA-NETO, U., AND PRINCE, J. L. 2002. Topology correction in brain cortex segmentation using a multi-scale, graph-based algorithm. *IEEE Trans. Med. Imaging* 21, 2, 109–121.

HATCHER, A. 2002. *Algebraic Topology*. Cambridge University Pres.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3d freeform design. In *SIGGRAPH*, 409–416.

JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. 2002. Dual contouring of hermite data. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 339–346.

JU, T. 2004. Robust repair of polygonal models. *ACM Trans. Graph.* 23, 3, 888–895.

KHO, Y., AND GARLAND, M. 2005. Sketching mesh deformations. In *SI3D*, 147–154.

KRIEGESKORTE, N., AND GOEBEL, R. 2001. An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical mr volumes. *Neuroimage* 14, 2 (August), 329–346.

MAMMEN, A. 1989. Transparency and antialiasing algorithms implemented with the virtual pixel maps technique. *IEEE Comput. Graph. Appl.* 9, 4, 43–55.

MANGIN, J.-F., FROUIN, V., BLOCH, I., REGIS, J., AND LOPEZ-KRAHE, J. 1995. From 3d magnetic resonance images to structural representations of the cortex topography using topology

preserving deformations. *Journal of Mathematical Imaging and Vision* 5, 297–318.

MATVEEV, S. 2003. *Algorithmic Topology and Classification of 3-Manifolds*. Springer-Verlag Berlin.

NEALEN, A., SORKINE, O., ALEXA, M., AND COHEN-OR, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph.* 24, 3, 1142–1147.

NOORUDDIN, F. S., AND TURK, G. 2003. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.* 9, 2, 191–205.

SHARF, A., LEWINER, T., SHAMIR, A., KOBELT, L., AND COHEN-OR, D. 2006. Competing fronts for coarse-to-fine surface reconstruction. In *Eurographics*, 389–398.

SHATTUCK, D. W., AND LEAHY, R. M. 2001. Automated graph based analysis and correction of cortical volume topology. *IEEE Trans. Med. Imaging* 20, 11, 1167–1177.

SZYMCZAK, A., AND VANDERHYDE, J. 2003. Extraction of topologically simple isosurfaces from volume datasets. In *IEEE Visualization*, 67–74.

WOOD, Z. J., HOPPE, H., DESBRUN, M., AND SCHRÖDER, P. 2004. Removing excess topology from isosurfaces. *ACM Trans. Graph.* 23, 2, 190–208.

ZHOU, Q.-Y., JU, T., AND HU, S.-M. 2007. Topology repair of solid models using skeletons. *IEEE Transactions on Visualization and Computer Graphics*, to appear.

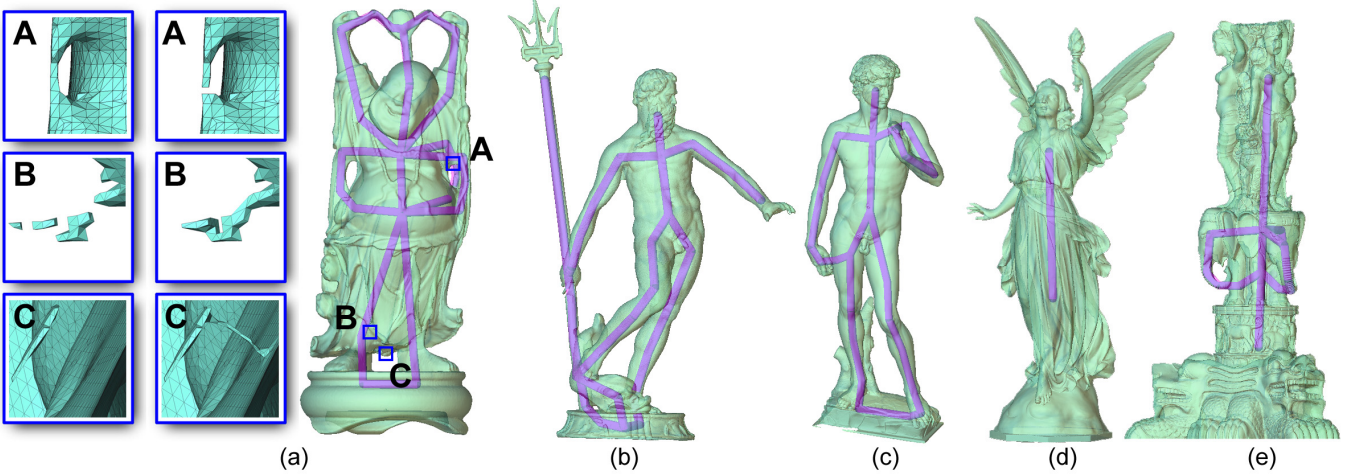


Figure 14: Topology repair of large scanned models (Buddha, Neptune, David, Lucy and the Thai Statue) by sketching. Close-up looks in (a) show the removal of a handle (A), the bridging of islands (B) and cavities (C) in the Buddha model.

Model	Input Polys	Ocree Depth	Ocree Leaf Cells	Topology Before (H/C/I)	Topology After (H/C/I)	Scan Conv. Time (sec)	Stage 1 Time (sec)	Stage 2 Time (sec)/ Steps	Contour Time (sec)	Memory (MB)	Output Polys
Figure 10 (a)	67904	7	76911	2 / 0 / 0	1 / 0 / 0	1.781	3.16	2.0 / 1	0.797	2	68284
Figure 10 (c)	67904	7	76911	2 / 0 / 0	1 / 0 / 0	1.781	3.14	2.0 / 1	0.797	2	68536
Figure 1	12184	7	14149	1 / 0 / 0	0 / 0 / 0	0.313	1.18	4.125 / 1	0.110	< 1	12882
Figure 12	25948	7	31376	8 / 0 / 0	14 / 0 / 0	0.64	1.05	10.05 / 3	0.265	< 1	26416
Figure 11	57500	7	66831	4 / 0 / 0	1 / 0 / 0	1.484	2.28	5.172 / 3	0.687	2	62700
Figure 9	53152	8	59691	5 / 0 / 5	1 / 0 / 0	1.375	2.25	36.687 / 9	0.562	2	54036
Figure 13	125152	8	139001	60 / 0 / 5	0 / 0 / 24	3.344	5.02	260.344 / 85	1.469	4	133882
Figure 14 (a)	1087716	10	3991052	11 / 17 / 7	6 / 0 / 0	43.75	131.094	117.875 / 29	47.719	87	3432358
Figure 14 (b)	6330753	11	8847449	60 / 5 / 38	3 / 0 / 0	181.329	289.594	634.516 / 100	110.78	193	7607896
Figure 14 (c)	8254150	12	34290104	21 / 72 / 6	3 / 0 / 0	317.766	1100.5	1426.859 / 96	444.89	748	29436068
Figure 14 (d)	28055742	12	46566011	0 / 0 / 1	0 / 0 / 0	947.515	1521.547	869.578 / 1	602.875	1015	39951458
Figure 14 (e)	10000000	12	66840790	4 / 0 / 0	3 / 0 / 0	454.296	2155.093	456.719 / 1	870.672	1456	57421186

Table 1: Performance results for examples in this paper. Timing is broken down into initial scan-conversion from meshes to volumes (using Polymender), the two stages of our guided topology editing algorithm (see Section 3.2, 3.3), and final contouring (excluding I/O, see Section 3.1). The steps are iterations in the second stage where an isolated element is removed. H, C, I stand for handles, cavities and islands.

A Proof of Proposition 2

Lemma 1 Let S be a FS of V , then \bar{V} is a FS of \bar{S} .

Proof: Consider a simple removal from V as $V' = V \setminus \{\delta, \sigma\}$. Taking the complement of both sides of identity yields $\bar{V} = \bar{V}' \setminus \{\widehat{\sigma}, \widehat{\delta}\}$. Note this is also a simple removal, since $D(\widehat{\sigma}) = D(\widehat{\delta}) + 1$, and $\widehat{\delta}$ is contained in only one element $\widehat{\sigma}$ in \bar{V}' (otherwise δ would contain some element other than σ that is not in V' , contradicting that V is a cell complex). Hence \bar{V} is a FS of \bar{V}' . Likewise, \bar{V} is a FS of \bar{S} , and can be reduced from \bar{S} by the dual of the simple removals that reduce V to S . \square

Lemma 2 Let S, T be FSs of V, \bar{V} , and $e \in S$ be an isolated element. Then then $T' = T \cup \widehat{W_{\bar{T}, S}(e)}$ is a FS of $\bar{V}' = \bar{V} \setminus W_{V, S}(e)$.

Proof: By Lemma 1, V is a FS of \bar{T} . Hence S is a FS of \bar{T} as well. By Proposition 1, $S' = S \setminus \{e\}$ is a FS of $\bar{T} \setminus W_{\bar{T}, S}(e) = \bar{T}'$. Since $W_{V, S}(e) \subset W_{\bar{T}, S}(e)$, V' can be reduced from \bar{T}' by the simple removals in $R_{\bar{T}', V'}$ that are not in $W_{\bar{T}, S}(e) \setminus W_{V, S}(e)$. Applying Lemma 1 again completes the proof. \square

Proposition 2 (See Section 3.3.2)

Proof: Observe that $\bar{V}' = \bar{V} \setminus W_{\bar{V}, T}(f)$. By Proposition 1, V' is a cell complex. Applying Lemma 2, $S^* = S \cup \widehat{W_{\bar{S}, T}(f)}$ is a FS of V' .

Hence it suffices to show that S' is a FS of S^* .

Let the unique paths associated with e and f be denoted as $R_e = \{e, \delta_e, \dots, \sigma_e\}$ and $R_f = \{f, \delta_f, \dots, \sigma_f\}$. We consider the sequence $R'_f = \{f, \delta_f, \dots, \sigma_f, \widehat{\sigma}_e, \dots, \widehat{\delta}_e\} \subset \bar{S}$. Since $\widehat{\sigma}_e$ contains σ_f , R'_f is a path associated with f and $R'_f \subset W_{\bar{S}, T}(f)$. We re-write R'_f as $\{f, \delta_1, \sigma_1, \dots, \delta_k, \sigma_k\}$ where $\{\delta_i, \sigma_i\}$ are simple removals. We show two properties of R'_f . First, elements in R'_f have alternating dimensionality. This is because $D(f) \leq D(\delta_1) - 1$ and $D(\sigma_i) = D(\delta_i) - 1$, yielding $D(f) \leq D(\sigma_k)$. On the other hand, since \widehat{e} contains $\widehat{\sigma}_k$ and by the definition of compatibility, we have $\widehat{\sigma}_k \leq D(\widehat{e}) - 1$ and $D(\widehat{e}) - 1 = D(f)$. All equalities are reached only when elements of R'_f have dimensionality $\{D(f), D(f) + 1, D(f), \dots, D(f) + 1, D(f)\}$. Second, δ_i contains no other element in the generating set $W_{\bar{S}, T}(f)$ except σ_i (for $i > 0$) and σ_{i-1} (for $i > 1$). Suppose δ_i contains another element $\tau \in W_{\bar{S}, T}(f)$, there would be a path associated with f of the form $\{f, \dots, \tau\}$ or $\{f, \dots, \tau, \tau'\}$. In either case, $\{f, \dots, \tau, \delta_i, \dots, \sigma_k\}$ or $\{f, \dots, \tau, \tau', \delta_i, \dots, \sigma_k\}$ is a different path than R'_f , contradicting the uniqueness of R_e and R_f . Based on the above discussion, it is not hard to show that the sequence of pairs $R = \{\{e, \delta_e\}, \dots, \{\sigma_e, \widehat{\sigma}_f\}, \dots, \{\widehat{\delta}_f, \widehat{f}\}\}$ (i.e., joining R_e with the dual of the reversed R_f and pairing consecutive elements) are simple removals from S^* . The remainder of S^* can then be reduced to S' by the dual of the simple removals in $W_{\bar{S}, T}(f)$ that do not belong to R . This completes the proof. \square