


Topology-Controlled Reconstruction from Partial Cross-Sections

Amani Shhadi¹ and Gill Barequet¹ 

¹Dept. of Computer Science, The Technion—Israel Inst. of Technology, Haifa 3200003, Israel
E-mail: amani.shhadi@campus.technion.ac.il, barequet@cs.technion.ac.il.

Abstract

The problem of 3-dimensional reconstruction from planar cross-sections arises in many fields, such as biomedical image analysis and geographical information systems. The problem has been studied extensively in the past 40 years. Each cross-section in the input contains multiple contours, where each contour divides the plane into different material types. The reconstructed object is a valid volume (surrounded by a closed surface) that interpolates the input slices.

Some previous works utilize prior information about the reconstructed object, such as its topology, for recovering the original shape of the object. These works assume that the input cross-sections are complete and do not contain areas of missing information. In many real-life cases, this assumption does not hold. Other existing works handle such inputs; however, the methods they suggest do not have topological guarantees for the reconstructed object.

In this work, we provide the first technique that provides topology control for 3-dimensional reconstruction from partial planar cross-sections. The input to our algorithm consists of an arbitrarily-oriented set of 2-dimensional cross-sections that may contain areas of missing information (“unknown” regions) and user-specified topology constraints on the reconstructed object. During the reconstruction process, we explore a set of distinct topologies for relabeling the “unknown” regions. We define a scoring function for calculating the likelihood of each topology. We then examine a set of representative topologies and choose the reconstruction that simultaneously satisfies the global topology and optimizes the scoring function.

1. Introduction

Reconstructing a 3-dimensional object from a given set of planar cross-sections has been widely studied since the pioneering work of Keppel [Kep75]. The problem arises in many fields, such as medical imaging, in which we try to reconstruct and visualize human organs using, for example, MRI, CT, or ultrasound scans. The input is preprocessed manually by contouring the organ boundary in the 2D sections (see Figure 1). These contours separate the “inside” and “outside” regions in each section. In a more general setting, different labels (or “colors”) are assigned to the different contours, for characterizing different material types (muscles, bones, blood vessels, etc.). In addition, sections may contain noisy or incomplete regions; such regions will be classified as “unknown.”

In many cases, we have a considerable amount of prior information about the sought object, that can help in the reconstruction process, such as the topology, which can be described by the number of connected components and their genus. The topology is a significant piece of helpful information that can be used for recovering the correct shape of the object. It can reduce the number of cross-sections needed for recovering accurately the desired object, and save the specialist’s time in the contouring process.

In this paper, we present a new topology-controlled reconstruction algorithm which handles the general case of cross-sections containing multiple contours of multiple colors and areas of miss-

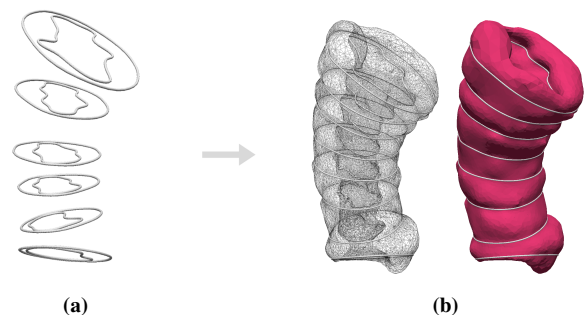


Figure 1: Input and output visualisation: (a) A set of non-parallel contours of a developing chicken heart from CT scans; (b) (right) The reconstructed three-dimensional chicken heart.

ing information, which we call “partial cross sections.” Given a set of such cross-sections, our algorithm reconstructs a 3-dimensional object that satisfies topology constraints specified by the user.

1.1. Three-Dimensional Reconstruction from Cross-Sections

The object-reconstruction problem was studied extensively in the past few decades. Early works in the field studied a simple version of the problem, trying to reconstruct a single object from parallel cross-sections while imposing different limitations, such as the number of contours in each slice, the geometries of the slices, and containment hierarchies of the contours [Kep75, FKU77, BPC*81, GD82, WA86, KdF88, KSdF88, FL99]. Other works addressed even a more fundamental problem, in which each input slice has a single contour [CP94, CS78, EPO91, MSS91, ZJH87]. These works failed to resolve complex inputs and generated unacceptable solutions with self-intersecting surfaces and gaps between the contours. More recent works tried to handle the more general case, in which no limitations were imposed on the contours in each slice [BG92, BCL96, BS96, OPC96, BGLSS04]; some other works tried to solve more general cases in which the input consisted of nonparallel cross-sections [BTS04, BM07, LBD*08]. A few other works studied the problem of reconstructing a three-dimensional object from multi-labeled contours [BVG11, BV09], where each contour was specified with a label that describes its type of material, and the goal was to reconstruct simultaneously geometrically-valid surfaces for all types of materials.

1.2. Reconstruction from Partial Cross-Sections

Most previous works assumed that the slices in the input were complete, *i.e.*, that no data were missing, and that the slices were segmented correctly. In fact, this assumption is not feasible in most real-life cases, since the input might in practice have noisy areas with incomplete information. Only a few previous works (*e.g.*, Refs. [BV09, BVG11]) considered partial cross-sections. To solve the problem, Barequet and Vaxman [BV09] removed the unknown areas from each cross-section, then computed the arrangement of the partial planes and its three-dimensional straight skeleton. Finally, they provided an extension to the algorithm of Liu *et al.* [LBD*08] for dealing with nonconvex cells in order to reconstruct the surface. Bermano *et al.* [BVG11] took an entirely different approach. Based on the inside and outside information given at each cutting plane, they used an indicator function for defining an implicit function on any cutting plane. This function was interpolated inside the cells defined by the cross-sections using mean-value transfinite interpolation. The indicator function was not continuous at the input slices, which led to ripples in the created surface. Thus, they had a post-processing step for smoothing their output. However, none of these works have topological guarantees for the reconstructed surface.

1.3. Topology-Controlled Reconstructions

Most previous works focused on interpolating a geometrically-valid object without considering the topological correctness of the reconstructed surface. These works made many topological errors, particularly when the cross-sections were not dense. Recent works [ZHCJ15, HZCJ17, LDK*18] introduced a topology-controlled reconstruction method in which the interpolated surface matched a user-specified genus in both two-labeled and multi-labeled scopes.

Given an arrangement of cross-sections that divides the space into polyhedral volumes (*cells*), Zou *et al.* [ZHCJ15] used a topology-controlled method which employed a divide-and-conquer strategy for exploring a family of distinct topologies of surfaces (so-called *tilings*) within each cell. Their method computed a score that evaluated the likelihood of each tile, then ran an optimization step that selected one tile topology per cell, such that 1. The obtained surface had the desired genus; and 2. The cumulative score of all tiles was maximized.

The scoring function of Zou *et al.* is based on a scalar indicator function defined within each cell, so that the level sets of the function interpolated the contours on the boundary of the cell. Then, using a bottom-up dynamic-programming technique, they chose one topology per cell, such that the union of the selected topologies met the two properties mentioned above. Huang *et al.* [HZCJ17] extended this algorithm to support multi-labeled domains.

None of the previous methods can handle partial inputs. Thus, our goal is to provide an algorithm for topology-controlled object reconstruction from partial multi-labeled cross-sections.

2. Problem Definition

We are given a set of planar cross-sections, each containing a set of closed non-intersecting curves. These curves split the cross-sections into “inside,” “outside,” and “unknown” regions (see Figure 2), where “Inside” indicates that this area is part of the reconstructed object; “Outside” indicates the area that does not belong to the object; and “Unknown” indicates that there is no reliable information about whether this area lies inside or outside the object.

The goal is to reconstruct a three-dimensional surface which is geometrically valid (*i.e.*, free of holes and intersecting loops) and interpolating the input curves. Furthermore, as mentioned above, in many cases, we have prior information about the topology of the original object. Thus, the number of connected components and genus can also be specified in the input. In such cases, the reconstructed object should also satisfy the given topology constraints.

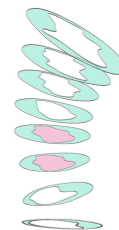


Figure 2: Input contours, “Inside” (green), “Outside” (white) and “Unknown” (pink) through a single point.

3. Our Contribution in a Glance

Topology control has been extensively studied in the past years for both two-labeled and multi-labeled domains. However, all previous works assume that the input is complete, *i.e.*, the entire cross-sections are available and segmented correctly. This assumption does not hold in most practical cases, where the input is noisy and contains unclassified regions.

This paper introduces a new topology-guided reconstruction algorithm which handles the more general problem, in which the input cross-sections may have missing information. We are not aware of any previous topology-constrained reconstruction algorithm that can handle such inputs. Our primary contribution is the reconstruction of a three-dimensional volume while effectively handling different forms of information at the boundary of the volume.

4. The Algorithm

The input to the algorithm consists of a set of planar cross-sections, which might contain areas labeled “unknown,” and a list of topology constraints specified by the number of connected components and the genus of each component that are sought in the output.

The output of the algorithm consists of a three-dimensional object, free of holes and intersections (*i.e.*, geometrically valid) with the user-specified genus, so that the object interpolates the labels specified at the given cross-sections.

The algorithm proceeds with the following steps:

1. Run a modified version of the multi-label reconstruction algorithm of Huang *et al.* [HZCJ17], while assigning the “unknown” material a new label. The output of this step is a 3-dimensional object which contains “unknown” regions (see Figure 3).
2. Relabel the unknown volumes so as to match the output with the topology constraints. This is done using a generalized version of the algorithm of Zou *et al.* [ZHCJ15] for handling three-dimensional inputs. This step consists of two main stages:
 - a. **Topology Exploration.** Explore the different topologies in each cell and score the likelihood of each topology.
 - b. **Topology Selection.** Choose one surface for each cell so that the combined surface meets the user constraints and the cumulative score is maximized.
3. Apply mesh fairing and refinement using the algorithm of Liu. *et al.* [LBD*08].

Our topology-controlled algorithm is the first to handle the case of multiple cross-sections, each one containing multiple contours of unrestricted geometries and containment hierarchies, and possibly with missing information in portions of any of the sections.

4.1. A Sketch of the algorithms of Zou and Huang

Both algorithms of Zou *et al.* and of Huang *et al.* are executed in two main stages: Enumeration and Selection.

The goal of the Enumeration step is to compute a set of topologically-distinct material interfaces for each cell, so as to interpolate the curve network on the boundary of the cell and to score the likelihood of each topology.

To accomplish this goal, Zou *et al.* use the level sets of an *indicator function* defined within the cell, adopting the *random-walk probability* as their indicator function for two-labeled domains. For multi-labeled domains, Huang *et al.* extend this idea to a vector function whose interface sets interpolate the boundary curve networks.

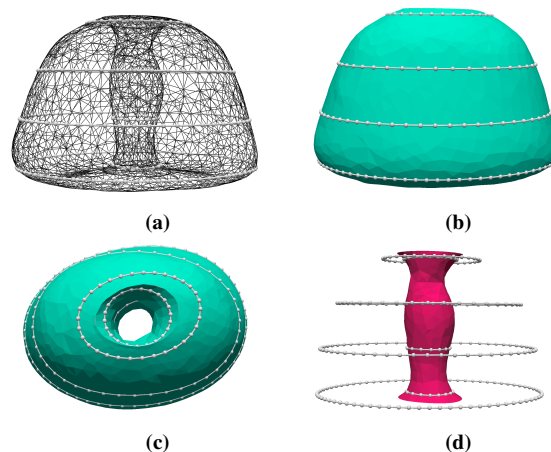


Figure 3: The output of the first step. (a) An overall view; (b,c) The reconstructed object without the unknown volume; (d) The reconstructed unknown volume.

In the Selection step, Zou *et al.* use a bottom-up dynamic-programming procedure, so that the final result matches the user-specified constraints, such that the cumulative score from all cells is maximized. Huang *et al.* use the same approach with minor changes in order to handle multiple materials simultaneously. In both works, the score is based on the indicator function.

4.2. Random Walk

In order to compute the random-walk function, we follow Zou *et al.* and build for each cell in the arrangement of planes an undirected graph $G(V, E)$ that describes a tetrahedral mesh, computed inside each cell so that the mesh agrees with the input contours.

We divide the set E into “seed” vertices and “non-seed” vertices. Vertices on the boundary of the cell are defined as seed vertices, and they are labeled with 1 if they lie inside the reconstructed object, and 0 otherwise. Vertices that lie inside the cell are defined as “non-seed” vertices, and no labels are associated with them.

Each edge $e_{i,j} \in E$, connecting vertices $i, j \in V$, is associated with a positive weight $w_{i,j}$ that represents the bias of the walker, when it stands at vertex i , to proceed to vertex j , among all neighbors of i . Zou *et al.* and Huang *et al.* defined $w_{i,j}$ to be constant for all edges, thereby making the choice of the next step to be random without any preference for any specific label. However, different weights can be assigned to edges in order to reflect a bias towards some direction. We review our weight function in the next section.

When the walker stands at vertex i , the probability that it will advance to a neighboring vertex j is set to $w_{i,j} / \sum_{k \in N(i)} w_{i,k}$, where $N(i)$ denotes the set of vertices that are neighbors of vertex i .

Finally, the random-walk function provides the probability x_i of a random walker, when it stands at a non-seed vertex i , to perform a walk that ends in a seed vertex labeled 1. As in Ref. [ZHCJ15, p. 128 (left), Eq. (1)], we set this probability to be the solution (for x_i) of the system of linear equations $x_i - \frac{\sum_{j \in N(i)} w_{i,j} x_j}{\sum_{j \in N(i)} w_{i,j}} = 0$. The level sets of the random-walk function define different tiles with various topologies for each cell.

4.3. Reconstructing the Unknown Volume

In contrast with the algorithms of Zou *et al.* and of Huang *et al.*, we handle the unknown volume the next step. Thus, given a cell with unknown areas in the cross-sections, we wish to prefer the unknown label upon other labels in order to control better the topology of the final object reconstructed in this step. To this aim, we define different weights for the edges of the graph in order to get a biased random walker. The weight of an edge $e_{i,j}$ is defined as

$$w_{i,j} = \frac{1}{\min(d(i), d(j)) + 2}, \quad (1)$$

where $d(i)$ represents the distance from a vertex i to the unknown areas in the cross-sections.

4.3.1. Scoring function

In addition, we want to affect the topology in which the algorithm labels the unknown volume. Thus, we prefer solutions in which the reconstructed unknown volumes connect a large number of labels in order to allow better control over the topology of the reconstructed object. To achieve that, we modify the scoring function accordingly. The function we used for scoring the likelihood of a tiling topology T is

$$S_T = |\{\ell \mid \exists v \in U_T : e_{u,v} \in G \text{ and } \ell_T(u) = \ell\}|, \quad (2)$$

where U_T is the unknown volumes in T , and $\ell_T(x)$ is the label at x as determined by T . This scoring function might result in less reasonable solutions, in which the unknown volume is connected to other pieces of material through a single point (see Figure 4). Therefore, we require that a vertex v have at least four neighbors of label ℓ in order to be considered in the set defined in Equation (2).

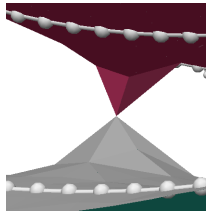


Figure 4: An unknown volume (pink) which is connected to another piece of material (gray) through a single point.

4.4. Labeling the Unknown Volume

We now describe the relabeling of the unknown volumes. Unlike previous methods, in which the input consists of only the sections, in our case it also contains three-dimensional *unknown volumes* (Figure 3d). The cross-sections partition the space into convex *cells* that include the unknown volumes. Our goal is to relabel these volumes so that the final reconstructed object will be geometrically valid and will satisfy the specified topology constraints (Figure 5). First, in the *exploration* step, we consider a family of distinct topologies by generating multiple tilings with different topologies within each cell of the unknown volumes. Then, in the *selection* step, we choose one tiling per cell, so that the combined surface meets the topology constraints and agrees with both input sections and the boundaries of the unknown volumes.

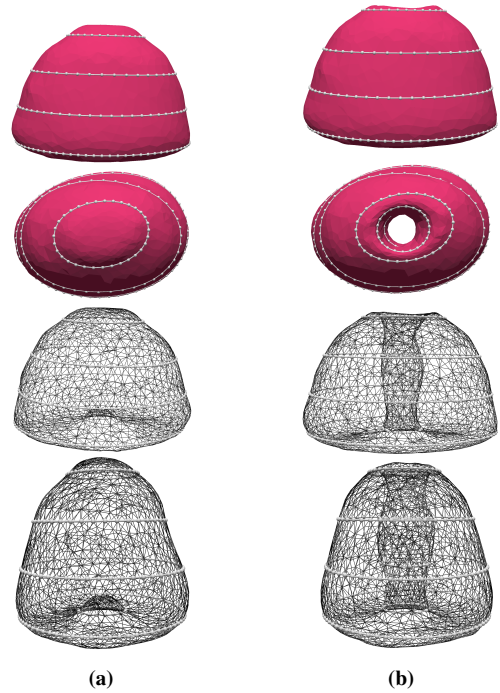


Figure 5: The output of the algorithm, applied to the example shown in Figure 3 from different viewpoints: (a) (left column) With a genus-0 constraint; (b) (right column) With a genus-1 constraint.

4.4.1. Topology exploration

The indicator function of Zou *et al.* handles cases in which the “known” values lie on the cross-sections only, and does not aim to deal with cases in which they lie between cross-sections. In our case, the values on the boundary of the unknown volume are calculated during the reconstruction of the unknown volumes. Thus, we extend the indicator function to handle bounded volumes in which the “known” values lie inside cells.

To achieve this goal, we modify the set of seed vertices in each cell to contain both the vertices on the cross-sections and the vertices of the unknown volume boundary (see Figure 6). We calculate the probability of a random walker that starts at a vertex v inside the unknown volume to end at a seed vertex, either at the cross-sections or the volume boundaries. Labels on the sections are assumed to be correct, *i.e.*, more reliable than labels inside the cell which were determined in the reconstruction of the unknown volume. For preferring labels on the sections, the weights of the biased random walker are as in Equation (1), however this time $d(i)$ is the distance from vertex i to the sections.

Similarly to Zou *et al.*, we use the level sets of the indicator function to define different tilings. However, unlike in their method, in which the level sets induce a partitioning of only the interiors of the cells and all tilings interpolate the fixed labels on the cross-sections, in our case, the level sets induce different partitions of the sections too. This makes the problem more challenging since neighboring cells might define different (contradicting) labeling for the same vertices (see Figure 7), which we need to avoid.

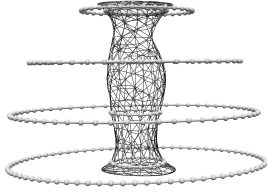


Figure 6: The boundary of the unknown volume.

In order to overcome this obstacle, we calculate the random-walk probabilities incrementally. First calculate in a random cell the values of the random-walk function. Then, we propagate the values on the cross-sections (that contain unknown regions) to the neighboring cells. For each neighboring cell c , we calculate the values of the random-walk and propagate the newly-calculated values on the cross-sections to the neighboring cells of c , and so on.



Figure 7: Neighboring cells with different labels of the vertices of the cross-sections. The upper cell defines the outer circle as “outside,” while the lower cell defines the inner circle as “outside.”

4.4.2. Scoring

We use the same function as Zou *et al.* and Huang *et al.* for scoring a tiling set. Our function is based on the random-walk probabilities. Given a tile T and the random-walk probabilities for each point x inside the cell, we denote by ℓ_T the label at x as determined by the tile T . The likelihood of the tile T at x is equal to

$$h_T(x) = \begin{cases} \log(f(x)) & \ell_T(x) \text{ is labeled “inside”}; \\ \log(1 - f(x)) & \ell_T(x) \text{ is labeled “outside”}. \end{cases}$$

Since the likelihood of each point is defined using the logarithm function, the joint likelihood of all the points inside the cell can be calculated by summation, and the likelihood over the entire cell space Ω is equal to $h_T(x) = \int_{\Omega} h_T(x) dx$.

4.4.3. Topology selection

In this step, our goal is to choose, for each cell that consists of “unknown” material, one tiling topology from those computed in the exploration step, so that the union of the cells classified as “unknown” with the other cells, computed in the step of reconstructing the unknown volume, will satisfy the user-specified genus and will have the maximum cumulative score among all possible choices.

Zou *et al.* and Huang *et al.* used a bottom-up dynamic-programming (DP) procedure for finding the optimal solution. Their solution might contain neighboring cells with a different level

set, which fits their settings, in which the values on the cross-sections are fixed, and, thus, the level sets inside the cells are independent. However, in our case, the cross-sections might contain “unknown” areas, that must be relabeled. That is, the values on the cross-sections are not fixed, and different level sets might define different values on the cross-sections (see Figure 8).

To solve the problem, we first divide the unknown cells into disjoint connected components of unknown cells. We mark the number of these components by m , and refer to these components as C_1, C_2, \dots, C_m . For each connected component C_i , we define the set of suggested level sets, L_{C_i} , to be $L_{C_i} = \bigcup_{c \in C_i} L(c)$, where $L(c)$ is the set of level sets generated by the cell c in the exploration step.

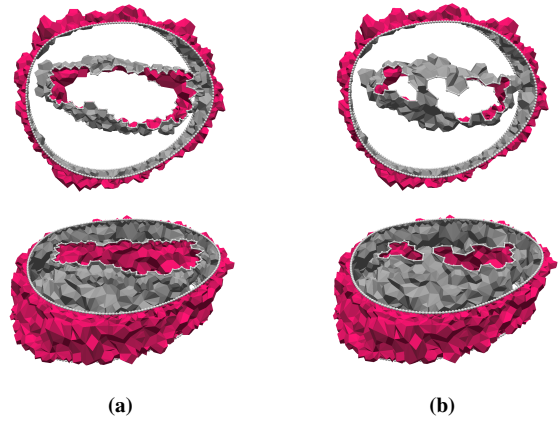


Figure 8: Different level sets of one cell (from different viewpoints): (a) With one loop on the cross-section; (b) With two loops on the cross-section.

Second, for each level set $\ell \in L_{C_i}$, we build the level set of the entire connected component C_i . The values of the border cross-sections of any connected component C_i do not contain unknown regions. Thus, they are not affected by the level sets. We merge the subsurface of C_i defined by the level set ℓ with the previously-calculated cells that do not contain unknown volumes and the other unknown volumes regardless of their level sets. For each level set, in each unknown connected component, we compute the score as mentioned above, and apply a dynamic-programming procedure to the unknown connected components for choosing the tiling that maximizes the score among all the tilings that satisfy the given topology constraints.

4.4.4. Reducing the number of level sets

Our DP procedure considers entire “unknown” connected components, a fact which reduces the number of considered topologies to at most half of those in the algorithm of Zou *et al.* and of Huang *et al.* (who consider individual cells). Still, in our implementation, the step of building the entire surface and calculating its genus is computationally expensive. In addition, the number of suggested level sets grows significantly with the number of cells in each unknown connected component since, unlike with Huang *et al.*, we allow the cells to contain surfaces with holes. Thus, we add more constraints in order to reduce the number of level sets.

First, we adopt their strategy of removing tilings that already have more connected components or a higher genus than the specified constraints. Second, for each unknown connected component, we calculate the Euler characteristics of the different level sets in L_C , and keep, for each value, only the tiling with the maximum score. Third, since we assume that the input contours cross each component inside the object at least once, we remove tilings that contain disconnected components inside the cell (see Figure 9).



Figure 9: A disconnected component inside a cell (from two different viewpoints).

4.5. Smoothing

Like Zou *et al.* and Huang *et al.*, we post-process the final shape using the method of Liu *et al.* [LBD*08], obtaining an overall smooth shape that interpolates the input curves (see Figure 10).

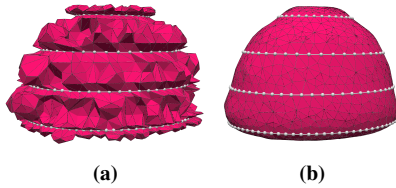


Figure 10: The output of the algorithm: (a) Before applying the smoothing procedure; (b) After applying the smoothing procedure.

5. Results

In fact, there are no other algorithms that combine object reconstruction from partial sections with topology constraints imposed on the output. Therefore, we cannot compare our algorithm directly to any previous algorithm. Nevertheless, we provide the results of some of the comprehensive experiments that we conducted.

We experimented with our algorithm on a few synthetic examples as well as on several nontrivial real examples, *e.g.*, a chicken heart and a hip bone. The input files were generated using the examples of Zou *et al.*, where in some cases we changed the materials of the input contours to be unknown, and in other cases, we added new contours with unknown materials. The outputs of our algorithm depended on the results we obtained by invoking the algorithm of Huang *et al.* Thus, we ran the algorithm of Huang *et al.* several times with different parameters.

5.1. Chicken Heart

In the chicken-heart case, we show three different examples. In the first two examples, we have two neighboring cross-sections with unknown regions (Figures 11 and 12), while in the third one, the two cross-sections are not neighbors (Figure 13).

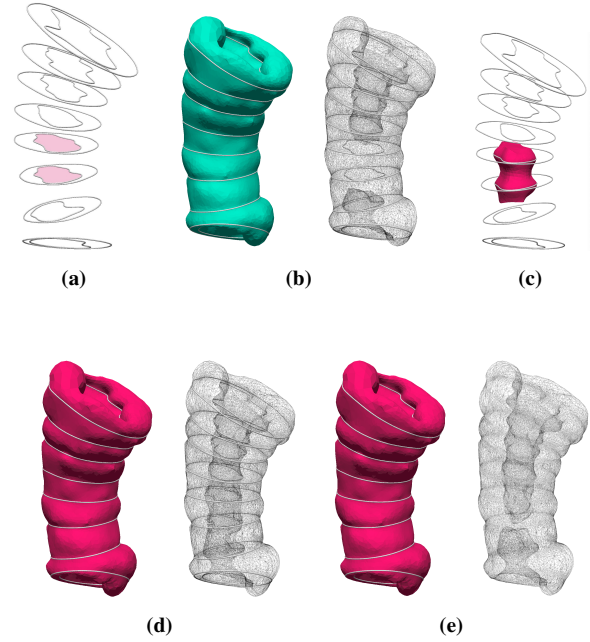


Figure 11: Applying the algorithm to a developing chicken heart with unknown regions. (a) The input to the algorithm contained two cells with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang *et al.* without the unknown material with a genus-1 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with a genus-1 constraint; (e) The output of our algorithm with a genus-0 constraint.

Figure 11a shows the input cross-sections in the first example. We ran the algorithm of Huang *et al.* with a genus-1 constraint. Figure 11b shows the reconstructed object without the unknown material, while Figure 11c shows the reconstructed unknown material. We ran our algorithm with both 1-genus (Figure 11d) and 0-genus (Figure 11e) constraints. The obtained results matched the specified constraints. In the 1-genus case, the algorithm relabeled the unknown region so that the reconstructed object contained one hole (Figure 11d), while in the 0-genus case, the algorithm reconstructed an object containing no holes (Figure 11e).

In the second example (Figure 12), both runs (of Huang *et al.* and our algorithm) were with a genus-0 constraint. Figure 12b shows the object reconstructed by the algorithm of Huang *et al.* without the unknown material, while Figure 12c shows the reconstructed unknown material. Figure 12d shows the result of our algorithm, which matched the user-specified genus (that is, no holes).

In the third example (Figure 13), the two cells that contained unknown regions were not neighbors. Thus, we obtained two unknown regions by running the algorithm of Huang *et al.* (Figure 13c). We ran our algorithm with both a 1-genus (Figure 13d) and a 0-genus (Figure 13e) constraint, and in both cases, our algorithm relabeled the unknown regions to match precisely the topology constraints. In the first case, we obtained one hole in the object, while in the second one, the object had no holes.

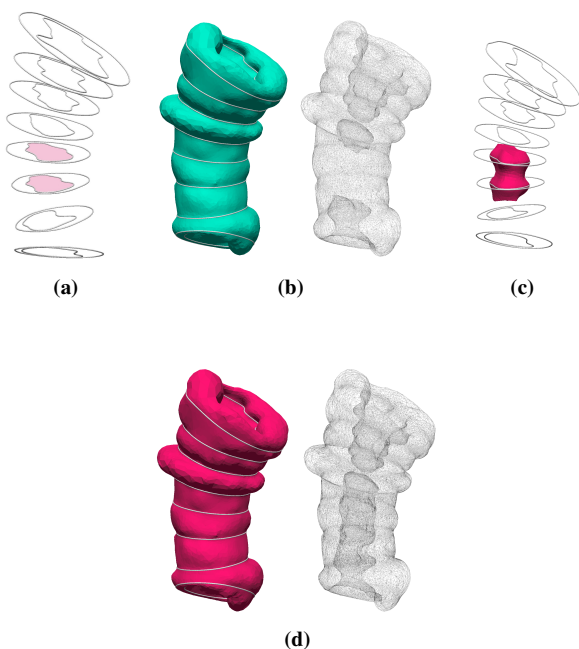


Figure 12: Applying the algorithm to a developing chicken heart with unknown regions. (a) The input to the algorithm contained two cells with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang *et al.* without the unknown material with a genus-0 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with a genus-0 constraint.

5.2. Hip Bone

In the case of the hip bone, we also present three examples. The first example contains one cross-section with an unknown region (Figure 14), while the two other examples contain two cross-sections with non-neighboring unknown materials (Figures 15 and 16).

Figure 14a shows the input of the first example. We ran the algorithm of Huang *et al.* with a genus-3 constraint. Figure 14b shows the reconstructed object without the unknown material, while the reconstructed unknown material is shown in Figure 14c. We ran our algorithm with a genus-2 constraint (Figure 14d) and a genus-3 constraint (Figure 14e). In the first case, the reconstructed object contained two holes. In the second one, it contained three holes; both results matched the user-specified constraints.

Figure 15 presents the second example. Figure 15a shows the input cross-sections with the unknown material. We ran the algorithm of Huang *et al.*, where the constraints were two connected components with a genus-0 constraint for one component, and a genus-1 constraint for the other component. Figure 15b shows the reconstructed objects without the unknown material, while Figure 15c shows the reconstructed unknown material. We ran our algorithm with two different constraints. In the first case, one connected component had a genus-2 constraint (Figure 15d), and in the other case, one connected component had a genus-1 constraint (Figure 15e). Both outputs of the algorithm matched the user-specified constraints. In the first case, the reconstructed object contained two holes, while it contained one hole in the other case.

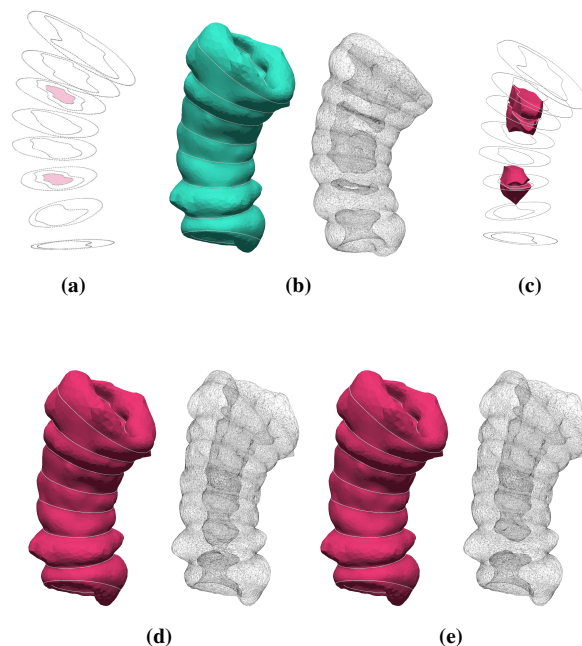


Figure 13: Applying the algorithm to a developing chicken heart with two unknown regions. (a) The input to the algorithm contained two cells with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang *et al.* without the unknown material with a genus-1 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with a genus-1 constraint; (e) The output of our algorithm with a genus-0 constraint.

The third example (Figure 16) shows how our algorithm satisfies topology constraints with multiple connected components. The input cross-sections are shown in Figure 16a. We ran both algorithms (the algorithm of Huang *et al.* and our algorithm) with the constraints of two connected components with a genus-0 constraint for one component, and a genus-1 constraint for the other component. Figure 16b shows the object reconstructed by the algorithm of Huang *et al.* without the unknown material, while Figure 16c shows the reconstructed unknown material. We obtained two components, one with no holes and the other with one hole (Figure 16d), which matched the topology constraints.

6. Implementation and Performance

We implemented our algorithm in C++ and tested its performance on a 4-core 2.4GHz PC with 8GB of main memory. The typical running time of our algorithm was affected mainly by the number of input sections and the numbers of vertices and tetrahedra. In addition, the running time was highly affected by the number of unknown components. Table 2 summarizes the running times on the examples discussed in Section 5. The running time was dominated by two steps: Minimizing the number of level sets, and selecting the topology. We noticed that the minimization of the number of level sets was very effective. The number of the level sets before and after this step for each unknown component are shown in Table 1.

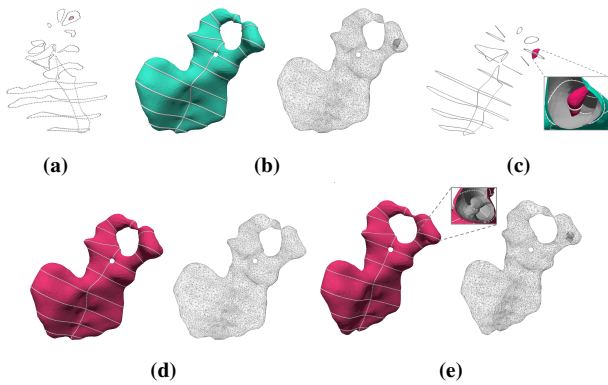


Figure 14: Running the algorithm on a hip. (a) The input contains one cell with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang et al. without the unknown material with a genus-3 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with a genus-2 constraint; (e) The output of our algorithm with a genus-3 constraint.

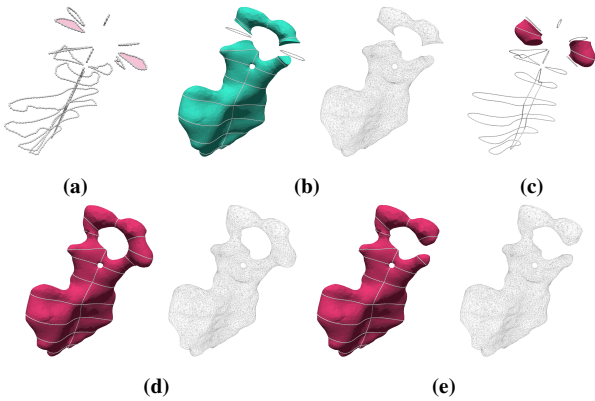


Figure 15: Running the algorithm on a hip. (a) The input contains two cells with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang et al. without the unknown material with two connected components, one with a genus-0 and the other with a genus-1 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with a genus-2 constraint; (e) The output of our algorithm with a genus-1 constraint.

7. Limitations of the Algorithm

Even though our algorithm reconstructs successfully complex real-life examples, it still has some limitations. First, it highly depends on the output of the algorithm of Huang *et al.*, and in some cases, our algorithm might fail in finding a solution with the user-specified topology constraints due to an inappropriate reconstruction of the unknown volume. To handle this problem, we tried to use different constraints in case the reconstruction process fails. However, in the future, we would like to seek a more efficient way.

Second, non-intuitive holes are sometimes the only possible way to satisfy the topology constraints. Figure 17 shows an example in which the algorithm reconstructed such a hole inside the unknown

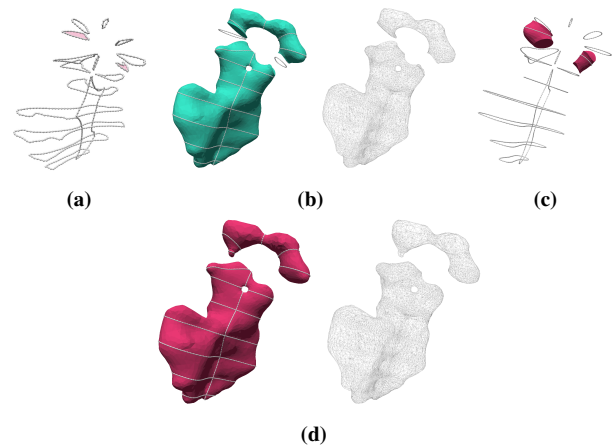


Figure 16: Running the algorithm on a hip. (a) The input contains two cells with unknown regions (pink); (b) The reconstructed object, using the algorithm of Huang et al. without the unknown material and with two connected components, one with a genus-0 and the other with a genus-1 constraint; (c) The reconstructed unknown material; (d) The output of our algorithm with two connected components, one with a genus-0 and the other with a genus-1 constraint.

volume. Usually, such an artificial hole is very small, unlike “real” holes that are reconstructed reasonably size.

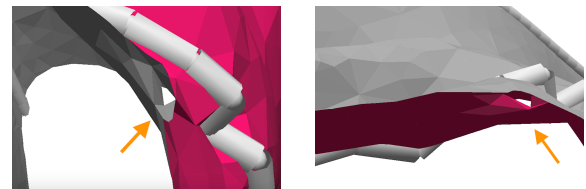


Figure 17: A small hole produced in inside the unknown volume.

8. Conclusion and Future Work

To the best of our knowledge, we present the first algorithm that provides topology control for three-dimensional object reconstruction from partial cross-sections. In its first step, our algorithm reconstructs the object while considering the unknown material as a special type of material. In the second step, our algorithm relabels the unknown regions so that the reconstructed object is topologically valid and it satisfies the user-specified topology constraints.

Since our algorithm uses the same approach as Huang *et al.*, which handles multi-labeled inputs, our algorithm can be generalized easily to handle multi-labeled inputs in which all labels inside the unknown volume appear on the unknown volume boundary.

In addition, as in Huang’s method, we can allow the user to steer the reconstruction process by choosing inside the unknown volume one topology per cell out of the list computed by our algorithm.

Table 1: A comparison of the number of level sets in each unknown component before and after running the minimization step.

Reducing the number of level sets				
Example	Figure	Unknown components	Initial no. of level sets	Minimized no. of level sets
Chicken Heart	11d	1	44	6
	11e			
	12d			
	13d	2	21	7
13e	16		8	
Hip Bone	14d	1	5	2
	14e			
	15d	2	5	2
	15e		2	2
	16d		1	1
		5	2	

Table 2: Running times of the algorithm.

Running Time				
Example	Figure	Tetrahedra	Vertices	Running Time (sec.)
Chicken Heart	11d	75,096	12,723	29.56
	11e			28.91
	12d			30.56
	13d	78,417	13,226	50.05
13e	50.21			
Hip Bone	14d	21,481	3,904	3.57
	14e			3.52
	15d	21,217	3,862	3.55
	15e			3.49
	16d			3.35

References

- [BCL96] BAJAJ C., COYLE E., LIN K.-N.: Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing* 58, 6 (1996), 524–543. 2
- [BG92] BOISSONNAT J.-D., GEIGER B.: Three dimensional reconstruction of complex shapes based on the delaunay triangulation. *Technical Report 1697, Inria-Sophia Antipolis, France* (1992). 2
- [BGLSS04] BAREQUET G., GOODRICH M. T., LEVI-STEINER A., STEINER D.: Contour interpolation by straight skeletons. *Graphical Models* 66, 4 (2004), 245–260. 2
- [BM07] BOISSONNAT J.-D., MEMARI P.: Shape reconstruction from unorganized cross-sections. *Symp. on Geometry Processing* (2007), 89–98. 2
- [BPC*81] BATNITZKY S., PRICE H., COOK P., COOK L., DWYER III S.: Three-dimensional computer reconstruction from surface contours for head ct. *J. of Computer Assisted Tomography* 5, 1 (1981), 60–67. 2
- [BS96] BAREQUET G., SHARIR M.: Piecewise-linear interpolation between polygonal slices. *Computer Vision and Image Understanding* 63, 2 (1996), 251–272. 2
- [BTS04] BOGUSH A., TUZIKOV A., SHEYNIN S.: 3d object reconstruction from non-parallel cross-sections. In *Proc. 17th Int. Conf. on Pattern Recognition* (Sep. 2004), vol. 3, pp. 542–545. 2
- [BV09] BAREQUET G., VAXMAN A.: Reconstruction of multi-label domains from partial planar cross-sections. *Computer Graphics Forum* 28, 5 (2009), 1327–1337. 2
- [BVG11] BERMANO A., VAXMAN A., GOTSMAN C.: Online reconstruction of 3d objects from arbitrary cross-sections. *ACM Trans. on Graphics* 30, 5 (2011), 1–11. 2
- [CP94] CHOI Y.-K., PARK K.: A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure. *The Visual Computer* 10, 7 (1994), 372–387. 2
- [CS78] CHRISTIANSEN H., SEDERBERG T.: Conversion of complex contour line definitions into polygonal element mosaics. *Computer Graphics* 12, 3 (1978), 187–192. 2
- [EPO91] EKOULE A., PEYRIN F., ODET C.: A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. on Graphics* 10, 2 (1991), 182–199. 2
- [FKU77] FUCHS H., KEDEM Z., USELTON S.: Optimal surface reconstruction from planar contours. *Comm. of the ACM* 20, 10 (1977), 693–702. 2
- [FL99] FIX J., LADNER R.: Multiresolution banded refinement to accelerate surface reconstruction from polygons. *Computational Geometry: Theory and Applications* 13, 1 (1999), 49–64. 2
- [GD82] GANAPATHY S., DENNEHY T.: A new general triangulation method for planar contours. *ACM Trans. on Computer Graphics* 16, 3 (1982), 69–75. 2
- [HZCJ17] HUANG Z., ZOU M., CARR N., JU T.: Topology-controlled reconstruction of multi-labelled domains from cross-sections. *ACM Trans. on Graphics* 36, 4 (2017), 1–12. 2, 3
- [KdF88] KEHTARNAVAZ N., DE FIGUEIREDO R.: A framework for surface reconstruction from 3d contours. *Computer Vision, Graphics and Image Processing* 42, 1 (1988), 32–47. 2
- [Kep75] KEPPEL E.: Approximating complex surfaces by triangulation of contour lines. *IBM J. of Research and Development* 19, 1 (1975), 2–11. 1, 2
- [KSdF88] KEHTARNAVAZ N., SIMAR L., DE FIGUEIREDO R.: A syntactic/semantic technique for surface reconstruction from cross-sectional contours. *Computer Vision, Graphics and Image Processing* 42, 3 (1988), 399–409. 2
- [LBD*08] LIU L., BAJAJ C., DEASY J., LOW D., JU T.: Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum* 27, 2 (2008), 155–163. 2, 3, 6
- [LDK*18] LAZAR R., DYM N., KUSHINSKY Y., HUANG Z., JU T., LIPMAN Y.: Robust optimization for topological surface reconstruction. *ACM Trans. on Graphics* 37, 4 (2018), 1–10. 2
- [MSS91] MEYERS D., SKINNER S., SLOAN K.: Surfaces from contours: The correspondence and branching problems. *Proc. Graphics Interface* 11, 3 (1991), 246–254. 2
- [OPC96] OLIVA J.-M., PERRIN M., COQUILLART S.: 3d reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoi diagram. *Computer Graphics Forum* 15, 3 (1996), 397–408. 2
- [WA86] WANG Y., AGGARWAL J.: Surface reconstruction and representation of 3-d scenes. *Pattern Recognition* 19, 3 (1986), 197–207. 2
- [ZHCJ15] ZOU M., HOLLOWAY M., CARR N., JU T.: Topology-constrained surface reconstruction from cross-sections. *ACM Trans. on Graphics* 34, 4 (2015), 1–10. 2, 3
- [ZJH87] ZYDA M., JONES A., HOGAN P.: Surface construction from planar contours. *Computers and Graphics* 11, 4 (1987), 393–408. 2