Eurographics Symposium on Geometry Processing 2009
Marc Alexa and Michael Kazhdan
(Guest Editors)

Volume 28 (2009), Number 5

Reconstruction of Multi-Label Domains from Partial Planar Cross-Sections

Gill Barequet[†] and Amir Vaxman[†]

Abstract

We present a novel algorithm for reconstructing a subdivision of the three-dimensional space (given arbitrarily-oriented slices of it) into labeled domains. The input to the algorithm is a collection of nonparallel planar cross-sections of an unknown object, where the sections might cover only portions of the supporting planes. (The information in the rest of these planes is, thus, "unknown.") Each cross-section consists of a partition of the plane into closed labeled ("colored") domains with no restrictions whatsoever on either their geometries or topologies, and without any assumptions about similarities between partitions of different sections. The problem is to reconstruct the original three-dimensional partition by interpolating simultaneously all the cross-sections, so that planar domains in the input are connected only to other domains of the same color, no two reconstructed spatial domains intersect, and no unnecessary gaps remain between the reconstructed colored domains.

The problem of reconstructing multiple-labeled domains arises, for example, in medical imaging, where different types of tissues are scanned and reconstructed at the same time. Partial slices are typical, for example, in ultrasound scanning. In this work we use the three-dimensional straight-skeleton of the arrangement of the cross-sections. Since the sections might be partial, cells of the arrangement might be nonconvex. For this we use the unambiguous definition, as well as the implementation of the computation, of the straight skeleton of a three-dimensional polyhedron that we presented in a recent work [BEGV08]. First, we define these cells and compute their skeleton. Second, we compute overlays of portions of sampled contours in the cross-sections, using the cell skeletons to guide the reconstruction of the mesh.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Reconstructing a three-dimensional object from a collection of its cross-sections has been a widely-investigated problem in the literature in the past thirty years. Most work on the subject deals with the interpolation between *parallel* sections, which comprise contours defining the geometry and topology of the intersections of the object with a series of (usually equally-spaced) parallel planes. In the majority of papers these contours are all of the same "color," that is, they describe the same type of material.

This reconstruction problem arises primarily in the fields of medical imaging, digitization of objects, and geographical information systems. Data obtained by medical imaging apparata, range sensors, or as elevation contours are interpolated in order to represent, reconstruct, and visualize human organs, CAD objects, or topographic terrains. It is assumed that a preprocessing step has already extracted from the raw data (usually a sequence of pixel images) the closed two-dimensional contours that delimit the material regions in each section. The goal is to compute a (usually triangulated) surface that tiles between these contours and forms a solid volume whose cross-sections at the given heights match the input slices.

Various algorithms for two-dimensional based triangulated surface reconstruction have been suggested in the literature (see, e.g., [BPCC81, FL98, FKU77, GD82, KD88, KSD88, Ke75, SP88, WA86, WW94]). Many early algorithms fail in complex instances (such as multiple branching), leave gaps between the contours, and/or generate un-

[†] Center for Graphics and Geometric Computing, Dept. of Computer Science, Technion—IIT, {avaxman|barequet}@cs.technion.ac.il

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

acceptable solutions (e.g., self-intersecting surfaces). Some algorithms [CP94, CS78, EPO91, MSS91, Sh81, ZJH87] reduce the more involved cases to the simple case where each slice contains only one contour. There have been several attempts [BCL96, BS96, Bo88, BGLS04, BG92, CD99, FO99, OPC96] to handle the interpolation problem without limiting the number of contours in the slices, their geometries, or their containment hierarchies.

A practical assumption in almost all previous work has been that the input consists of *parallel* slices, and, in addition, that adjacent layers are independent. Thus, only a single pair of successive parallel slices are considered and interpolated at each instance, and the reconstructed object is the concatenation of the interpolating models computed for all the layers. The only studies, that we are aware of, which handle input with nonparallel cross-sections, are [BTS04, Bm07, DP97, LBD+08, PT94].

Another practical assumption, which appears in almost all previous works, is that all the contours are "unicolored," that is, they describe the same type of material. Thus, there are no constraints on which contours should or should not be interpolated with other contours. This is a natural assumption, for example, when one reconstructs a single organ (e.g., bone) from medical data. However, input exists in which the cross-sections simultaneously describe several types of tissues (e.g., muscles, fat, bone, blood, etc.), so that each contour is given a *label* ("color") that characterizes its type of material. For such input, reconstruction of each label regardless of the other labels might result in inconsistency and intersections between the reconstructed outputs. The only studies, that we are aware of, which address multi-colored cross-sections, are [JWC+05, LBD+08]. These papers also suggested the term "curve network" to denote the partition of a planar region into colored subregions.

Thus, to the best of our knowledge, only Liu et al. [LBD+08] suggested an algorithm that interpolates between multi-colored curve networks in nonparallel cross-sections. Liu et al. considered the arrangement of the planes supporting the sections, and handled each cell of this arrangement separately. In every cell of the arrangement, they projected the portions of the contours that lie on the boundary of the cell onto the medial axis of the cell. By erecting "walls" connecting the original contour portions and their projections on the skeletons, they reconstructed portions of the colored surfaces (boundaries of the sought volumes) within each cell. Finally, they corrected jagged surfaces by general mesh-smoothing techniques that are independent of the reconstruction process.

Finally, all previous work assumed that all slices are *complete*, in the sense that no data are missing within the sections. This assumption is not practical, for example, for data originating from ultrasound scanning. Such areas, (within the input sections) where data are not available, can either belong or not belong to the reconstructed object, or, in the multi-label case, can belong or not belong to any tissue type, at all. We are not aware of any previous reconstruction algorithm that is able to handle such partial-slice situations.

The algorithm suggested in the current paper attempts to solve the slice-interpolation problem in its full generality. It handles parallel, nonparallel, or any combination of cross-sections. It supports the multi-label case, with no limit on the number of different labels. It can also handle the situation of partial sections, where data may be missing in portions of the sections. Naturally, within each section, the algorithm has no restrictions whatsoever on either the geometries or topologies of the labeled domains; further, it does not make any assumptions about similarities between partitions of different sections. The algorithm reconstructs the original three-dimensional partition by interpolating simultaneously all the cross-sections, so that planar domains are connected only to other domains with the same label, no two reconstructed spatial domains intersect, and no unnecessary gaps remain between the reconstructed labeled domains. The algorithm is guaranteed to interpolate a valid spatial partition for any possible input, and is intuitive in the sense that it tends *in practice* to minimize the surface area of the reconstruction. This is because it uses an offset distance function to locally decide which contour features to bind.

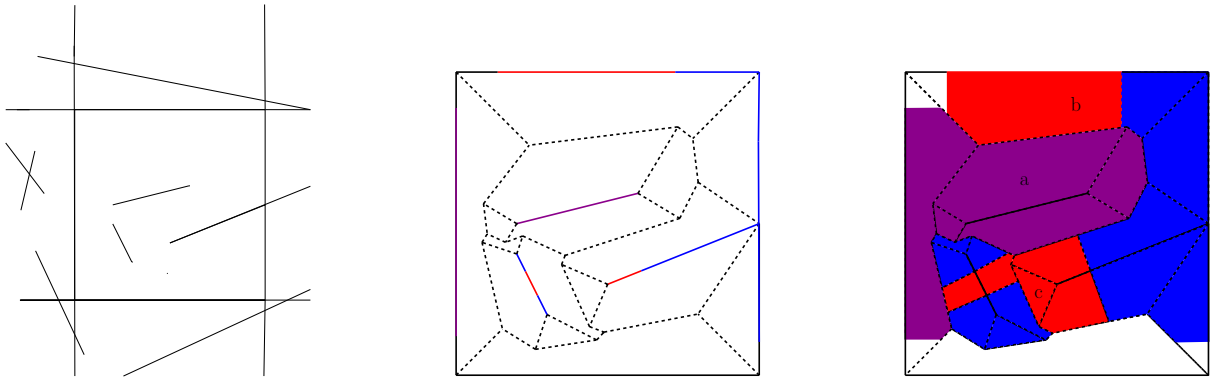
1.1. Overview of our Method

The input to our algorithm is a set of planar cross-sections, partitioned into a set of closed labeled ("colored") domains. Some planes might contain areas with an "unknown" color; these areas are treated as "holes" in the plane. The output is a colored triangulated mesh that represents the boundaries of the different domains in a partition of the space, such that this partition interpolates the known portions of the input planar cross-sections.

To distinguish between the arrangement of the planar cross-sections, and the labeled domains, we denote the latter as "images."

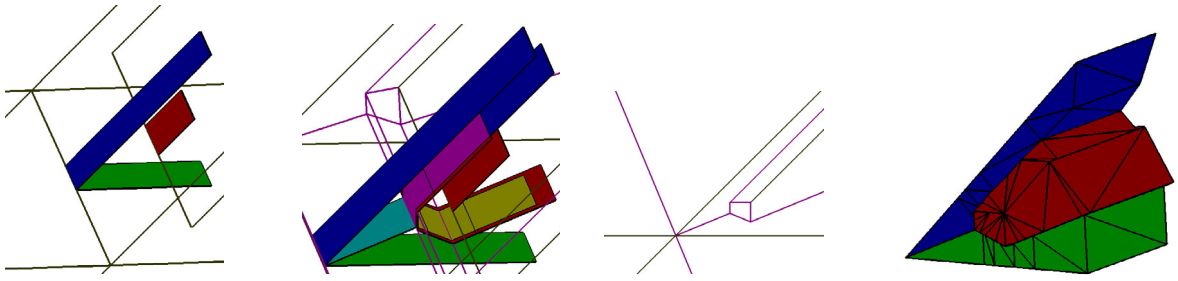
Our algorithm proceeds as follows (see Figure 2, and a two-dimensional exemplification of the algorithm in Figure 1):

1. Construct the spatial arrangement of the partial cross-sections (portions of planes).
2. Compute the three-dimensional straight skeletons of every cell of the arrangement of cross-sections; the skeletal cells are a refinement of the arrangement cells [BEGV08], such that every skeletal cell is defined by one arrangement cell (denoted the "base face").
3. For each skeletal cell of each arrangement cell:
   a. Project the images from every base face onto the other faces of that skeletal cell;
   b. Each skeletal face is shared by two skeletal cells on both sides. For each such skeletal face, compute the



(a) An arrangement of partial segments (b) The straight skeleton of a single cell with images (c) The reconstructed mesh

Figure 1: *An overview of the algorithm in 2D. The arrangement of partial segments is constructed. The straight skeleton of a single cell is computed (dashed). Notice that singular edges evolve as degenerate rectangles. A skeletal cell of type 'a' has obtuse walls, and therefore, the purple image stretches onto these walls (see Section 4). A cell of type 'b' does not have walls, and so the image of the base edge simply projects onto it. A cell of type 'c' is created from a degenerate vertex (which matches degenerate edges in 3D), and the image adjacent to this vertex projects onto the roof of the cell.*



(a) Three slices With projections Side view (b) 3D straight skeleton and projection (c) Full mesh

Figure 2: *Reconstruction from partial multi-colored slices. Two slices (with green and blue contours) are full, while the slice with the red contour lies only on a halfplane. The straight skeleton of the arrangement of partial planes is computed, the contours are projected, and, finally, the meshes are created accordingly.*

overlay of the projected images on both sides. For each such overlay, compute a constrained Delaunay triangulation;

c. Reconstruct portions of the mesh from these triangulated overlays.

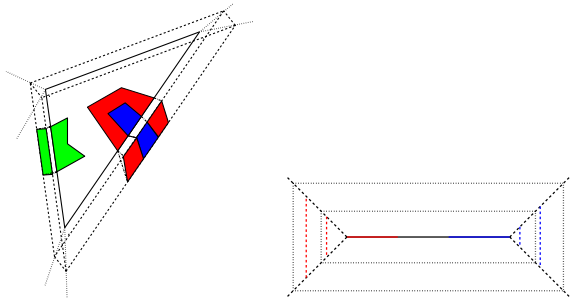
4. Stitch together all the pieces of the mesh in a consistent orientation.
5. Apply mesh fairing and refinement.

We now describe in detail the algorithm steps.

2. An Arrangement of Partial Cross-Sections

Using the arrangement of arbitrary-oriented planar cross-sections is a recognized technique for reconstruction of an object from nonparallel cross-sections (see, e.g., [Bm07, LBD+08]). Computing the arrangement of full planes is a topic well-studied in the literature. The cells of such an arrangement are all convex. In our setting, sections may be *partial*, so that the intermediate problem with which we deal is the computation of the arrangement of *portions* of planes. The cells of such an arrangement are not necessarily convex, and may contain degenerate or even disconnected planar patches "floating" inside the cell. We use the general Nef polyhedra [GHH+03], provided by [CGAL] to construct such an arrangement.

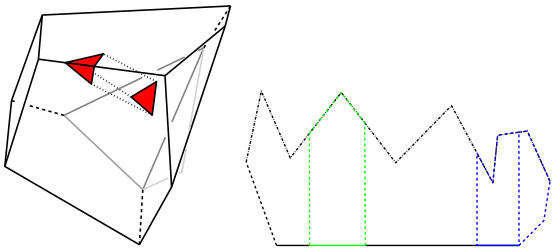
For practical purposes, in order to avoid infinite cells, we "wrap" the meaningful region of the arrangement (where there are data) with a large-enough bounding box. The images within the cross-sections are subdivided by the faces of the arrangement. The resulting arrangement contains (not



(a) Propagating of a degenerate patch

(b) Propagation of a degenerate segment in 2D

Figure 3: *The propagation of a planar patch (solid) with an image adjacent to the degenerate edges. The patch propagates into a triangular prism, and the image edges, which are adjacent to the degenerate edges, propagate into images faces. A similar example is depicted in the 2D setting, with a degenerate segment that propagtes into a rectangle, and the images adjacent at the vertices propagating into image edges.*



(a) A 3D cell

(b) A 2D cell. A green image is projected onto the roof, and a blue image is stretched onto the wall before being projected as well.

Figure 4: *A typical cell with obtuse walls is shown in the left image. The base (dark gray) has one triangular image (red), which is projected onto the roof (solid black). All the walls form obtuse angles with the base. (One neighboring cell, which forms an obtuse angle with the base, is shown in light gray.) In the left image a 2D depication of a skeletal cell is shown. The base is solid, the walls are dashed, and the roof is dashed-dotted.*

necessarily convex) cells, on whose faces portions of the original images of different colors lie.

From this point up to Section 6.1 we discuss the reconstruction of the mesh (portions of the boundaries of the reconstructed colored objects) within a single arrangement cell. We ignore arrangement cells of two types: (a) Cells that are empty of images; and (b) Cells that do not contain any image vertex or edge. A cell of the former type is free of any material, while a cell of the latter type is fully contained in a single domain. In both cases, such a cell does not contribute any portion of the reconstructed boundary mesh. In a post-processing step, we simply "glue" together all the interpolated pieces of mesh reconstructed within all the cells of the arrangement. Since the algorithm is interpolatory, that is, it produces colored meshes that are connected exactly to the images on the boundaries of the cells, and each image appears identically twice on twin faces, shared by two neighboring cells of the arrangement, the validity of the gluing step is guaranteed.

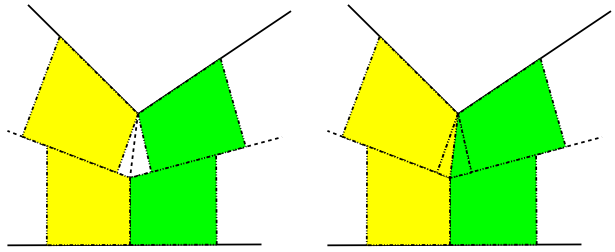
3. Straight Skeletons of Nonconvex Cells

It is well-known that the cells of the arrangement of full planes are always convex polyhedra. Recent works [Bm07, LBD+08] used the medial axis of a convex polyhedron in order to subdivide the cell, project on this axis the images lying on the boundary of the cell, and match portions of images accordingly. However, the cells of *partial* planes, supporting partial cross-sections, are not necessarily convex, and therefore, may have nonlinear medial axes, which complicates projective methods considerably. In order to avoid these complications, we use the three-dimensional straight-skeleton transformation (of a nonconvex polyhedron).

3.1. The Three-dimensional Straight Skeleton

The three-dimensional straight skeleton, recently defined unambiguously [BEGV08] for the first time, is a generalization of the well-known two-dimensional straight skeleton. The straight skeleton of a simple polyhedra (w/holes) is defined by a propagation process, in which all faces of the polyhedron propagate inward at equal rate, until the polyhedron vanishes. During this process, every face of the polyhedron sweeps a unique connected volumetric region inside the polyhedron (denoted a *3-cell* in [BEGV08]). It is easy to show that these 3-cells are monotone with respect to their original sweeping face using a similar argument made for the two-dimensional case [AA96]. The medial axis and the straight skeleton identify for convex polyhedra.

Every skeletal cell, induced by a base face of the arrangement of planes, contains skeletal faces of two more types: skeletal faces that form an obtuse dihedral angle with the base, and are denoted as "walls," and faces that form an acute angle with the base, which are called the "roof" of the skeletal cell (see Figure 4 for an illustration). The monotonicity property of skeletal cells determines that both the entire roof and entire wall are connected. That is, every wall (similarly, roof) face is connected to any other wall face by a path, which can only pass through wall faces. Note that neither all wall faces nor all roof faces must be adjacent to the base face. In a convex skeletal cell, which is the only case in an arrangement of full planes, all angles are acute, and therefore, such cells have no walls.



(a) Orthogonal projection (b) Wall stretching

Figure 5: *A 2D drawing of wall stretching. A simple orthogonal projection creates gaps in the surface although the images are continuous. Stretching the images on the obtuse walls fixes this effect.*

To support degenerate situations created by partial slices (in which portions of both sides of a plane belong to the same cell), we expand the definition to include the propagations of portions of planes and planar patches as follows. Every edge of the cell, which is adjacent to two faces of the cell lying on opposite sides of the *same* plane, propagates into a new face orthogonal to this plane. (This is a generalization of the definition of the straight skeleton of a straight-line graph [AA96], in which each segment is modeled as a degenerate rectangle.) Following this definition, we consider every floating planar patch as a degenerate prism, all of whose isolated edges (adjacent to only two faces of the arrangement) induce new faces as described above. Figure 3 illustrates these situations. Having computed the straight-skeleton for each arrangement cell, we obtain a partition of the arrangement cell into skeletal cells, such that every face (or a degenerate edge) of the arrangement is associated with such a skeletal cell, for which the arrangement face is the "base face." Each skeletal face, other than the base, has a counterpart face belonging to a neighboring skeletal cell.

4. Projecting Images onto Skeletal Faces

4.1. Extending Images onto Skeletal Walls

We proceed with projecting the images, lying on a a single base face f, onto the faces of the skeletal cell induced by f. Previous methods that applied a similar approach (e.g., [JWC+05]) used orthogonal projection onto the faces of the skeleton. However, this method works well only for an arrangement of full planes, in which the cells of the arrangement are convex, and hence the skeletal cells and faces are also convex (i.e., there are no walls). Consider a case in which the dihedral angle between two arrangement faces f_1, f_2 is reflex, resulting in an obtuse skeletal wall between them. In this case, a simple orthogonal projection of the images on both faces creates unnatural gaps in the resulting mesh, even if the images are continuous (see Figure 5).

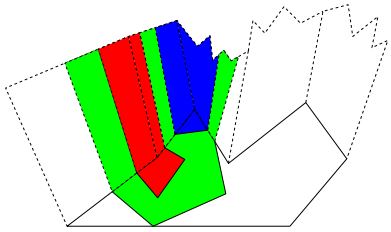
The monotonicity of the three-dimensional skeletal cell guarantees that any ray originating at either the base or the walls, directed orthogonally to the base and toward the inside of the cell, intersects exactly one point of the roof. To solve the gap issue, we extend the image edges that lie on the faces of the base onto the walls, "stretching" them upward until they reach the boundary of the roof. The vertices of the edges stretch across piecewise-linear routes through the faces of the walls, chosen in the following manner: for each face of the wall, if any image vertex is at the bottom of a wall edge, it is stretched along this edge. Otherwise, the vertex is stretched along a steepest ascent line inside this wall face. This stretching method has two appealing properties:

- No two routes cross each other, as all steepest-ascent lines are parallel, because the wall faces are planar. Occasionally, two routes may unite, if they both reach a common wall vertex, which means not all images may be stretched up to the boundary of the roof.
- If every point on the boundary of the face base is on an image edge, then every point on the wall is on an image. In other words, if there are no image gaps on the boundary of the base face, there are no image gaps on the walls. It is easy to see that the stretching of the edges propagates as a watertight front, which sweeps the entire wall. The stretching process is exemplified in Figure 6. The stretching process can be considered a morphing process from the boundary of the base to the boundary of the roof.

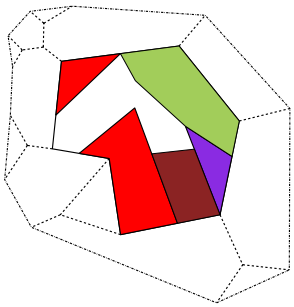
Having stretched the image, we can now project the image orthogonally onto the roof from both the base and the wall faces. When there are no walls in a skeletal cell, which is always the case where arrangement cells are convex, the boundary of the roof is exactly the base boundary, and so we simply project the image orthogonally from the base, thus conforming in this case to the projection method of [JWC+05].

4.2. Projection from Degenerate Edges

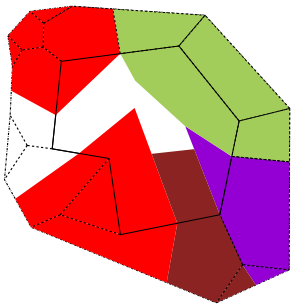
As defined in Section 3, degenerate edges of the arrangement propagate into faces orthogonal to the faces adjacent to this degenerate edge. Therefore, this edge is a degenerate base face of its skeletal cell. To properly produce the effect of skeletal propagation on the reconstruction, every image edge on a degenerate edge is also considered a degenerate image face on the propagating base. Thus, if every point on a degenerate edge is on an image edge, then every point on the propagating base is also on an image edge, and no new gaps are created. With the no-gap property of the wall-stretching step, and the property of degenerate propagation of images, we conclude that when every point on an arrangement face is on an image, then every point on the respective skeletal cell is on an image, original, stretched, or projected. For such an example, see Figure 7.



(a) Stretching an image from the edges of the base onto walls.

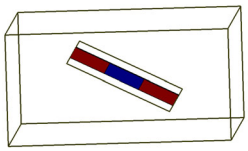


(b) Upper view (relative to base) of a base (solid) with several images adjacent at its edges, the surrounding walls (dashed), and the boundary of the roof (dashed-dotted).

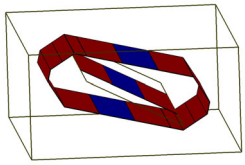


(c) Stretching the images in (b) onto the walls.

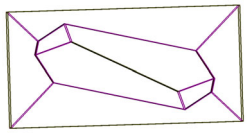
Figure 6: *Exemplifying the stretching of images onto walls.*



(a) The patch



(b) The overlay



(c) The side of the skeleton

Figure 7: *The propagation of an image from a floating patch inside a box. Notice how the red material propagates from the degenerate edges, which propagate into faces, and stretches to envelope the entire patch.*

5. Mesh Reconstruction

5.1. Image Overlay

At this stage, every skeletal face contains portions of images, either projected or stretched onto it from the original images on the base faces of the arrangement of planes. In fact, every skeletal face contains *two* sets of such images, because every such face has a twin skeletal face that is shared by a neighboring skeletal cell. We proceed by computing the two-dimensional *overlay* of these two partitions of the skeletal face. In the overlay, every face is associated with one of the following: two identical colors, two different colors, a color and the null color ("air"), or with the null color on both sides. As in the reconstruction algorithm of [BGLS04], we ignore the uniform cases (same color or null color), and compute the constrained Delaunay triangulation of overlay faces originating from images of two different colors (including a color and the null color).

5.2. Mesh Creation

We generalize the "spatial lifting-up" method of [BGLS04]. Every overlay of the projected images (described in the previous section) is associated with three planes: The overlay plane (the one that supports the skeletal face), denoted as the "middle" plane, and the two planes supporting the bases of the respective skeletal cells (which are also the planes supporting the respective cross-sections), denoted as the "upper" and "lower" planes (arbitrarily selected). The reconstructed mesh connects the images in the upper and lower planes, through the middle plane, within the two respective skeletal cells. The reconstruction of the portion of the mesh induced by the skeletal face has two steps. The first step is the creation of triangles in the overlay, as discussed in the previous section. The second step is the creation of the mesh triangles associated with this overlay, in a similar manner to that of [LBD+08]: two back-to-back mesh triangles are created per triangle in the overlay plane (in the case when the overlay is between a color and the air, only one triangle is created), and mesh triangles are erected between portions of original, or stretched images and their projection in the overlay.

After computing the mesh portion for every such overlay in each skeletal face within every skeletal cell, and within every cell of the arrangement of planes, we obtain a full consistently-orientated portion of the mesh in these cells, in which the boundary of every domain (color) is an oriented

2-manifold, and is adjacent, via back-to-back triangles, to other domains (or to the "air"). By gluing the reconstructed portions of mesh in all the arrangement cells, we obtain the fully reconstructed partition of space into colored domains.

5.3. Correctness

First, we claim that the boundary of every reconstructed spatial domain is a closed 2-manifold, and no two such boundaries intersect each other. This follows from the fact that every face of a section image, which is the result of intersecting a spatial domain with a plane, is projected onto both sides of the supporting plane of this section. On either side, this image face is overlaying either another domain of the same color, in which case the two are eventually connected, or another domain of a different color, in which case triangles that close up the boundary between the domains are created. Therefore, any domain is closed.

Every reconstructed domain is also a 2-manifold, resulting from the construction of the triangles for every overlay face. Mesh triangles are either on the overlay plane, or between an original edge and its projection, and in any case have only one adjacent triangle of the same triangle of the same color. Within one skeletal cell of the arrangement of planes, there are no intersections between the domains, because of the proper stretching step, and due to the orthogonal projection in the monotone skeletal cell. Since the creation of mesh portions within one cell is completely independent of the reconstruction within other cells, and the fact that the mesh portion lies completely inside the cell, no two reconstructions of mesh portions of two cells interfere with each other.

Second, we argue why no gaps (portions of space associated with the null color) are created inside a cell of the arrangement of planes, whose boundary is completely covered by non-null colors. When all the faces of a cell of the arrangement of planes are covered by images, none of which is associated with the null color, the images are stretched along the obtuse walls so as to fully cover them, as proved in Section 4. Therefore, the orthogonal projection always covers the entire cells of the three-dimensional skeleton. Every full overlay (having no null color faces) creates two portions of back-to-back surfaces without gaps between them. Therefore, every skeletal cell and, consequently, every arrangement cell with full images, do not contain any gaps.

6. Optional Improvements

6.1. Mesh Refinement

Our algorithm produces consistent meshes. However, because of the projective method, and creating a triangulation in the overlay plane, the mesh may exhibit sharp angles and long, skinny triangles, which denote a poor mesh. We follow the coarse-to-fine method established in [LBD+08], following the remeshing method of [Lp03], and the mesh smoothing method of [KCVS98]. We use similar smoothing parameters as [LBD+08].

7. User-Guided Reconstruction

An inherent shortcoming of all projective reconstruction algorithms is that the correspondence between images on planes is mostly dictated by the topology of the projective planes. This results in the image shape being of concern only on overlays of corresponding planes. In order to alter a reconstruction of full planes, one has to introduce new full cross-sections. Our algorithm has the benefit of working with all settings of piecewise-linear cross-sections. Thus, we allow the user to insert guiding patches of arbitrary topology inside arrangement cells, in order to alter the reconstruction, without having to re-sample a new full cross-section. Figure 8 depicts such an intervention.

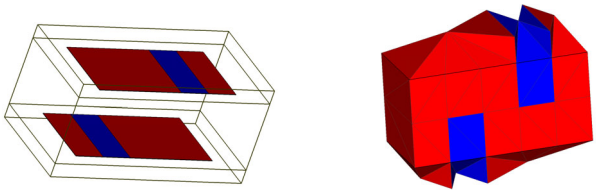
8. Experimental Results

Finally, we provide a few examples of running our algorithm on some synthetic and real input.

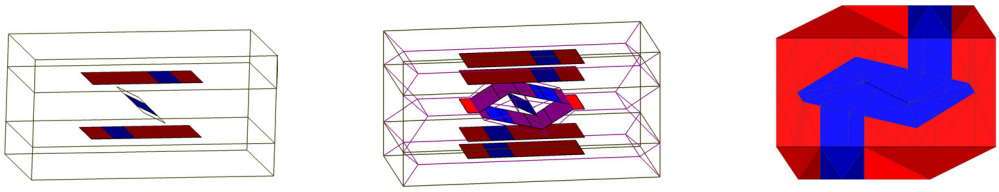
Figure 9(a) shows three floating patches (a section of a "3-knuckle," consisting of contours of the same color (brown). However, each is contained in a region of a different color. Figure 9(b) shows the different image overlays. Figures 9(c,d) show the reconstruction of one pipe and a 3-tunnel, respectively. Figure 9(e) shows the full reconstruction of the object.

Figure 10 demonstrates the effect of partial slices. The input is seen in (a): one full slice (at the top), containing a green contour surrounded by a blue region. This pattern is also reflected in two partial slices (halfplanes sharing their borderlines) at the bottom. In addition, on the two horizontal sides there are two partial slices containing red contours. Although the two sections with the red patches seem to "interfere" with the reconstruction of blue/green domains, this is not the case since the former sections are partial. (Note how the projection from partial slices, streched over the obtuse walls, looks like the green circle stretched on the vertical wall, obtuse to its defining faces.) This is manifested in (b), which shows the interaction of the overlays. The reconstructed domains are shown in (c,d,e). It is clearly seen how the green contours connect to each other.

Figure 11 shows the reconstructions of two sparsely-sampled binary meshes of a kitten and an elephant, which exhibit solutions to branching cases in nonparallel sampling. Figure 12 exhibits the reconstruction of a set of 13 domains, created by the intersection of four spheres. Figure 13 exhibits the reconstruction of the head of a cat, segmented into four domains, from slices, some of which contain only partial data. Table 1 summarizes the algorithm's statistics.



(a) Two slices with a similar blue stripe (b) The reconstructed mesh

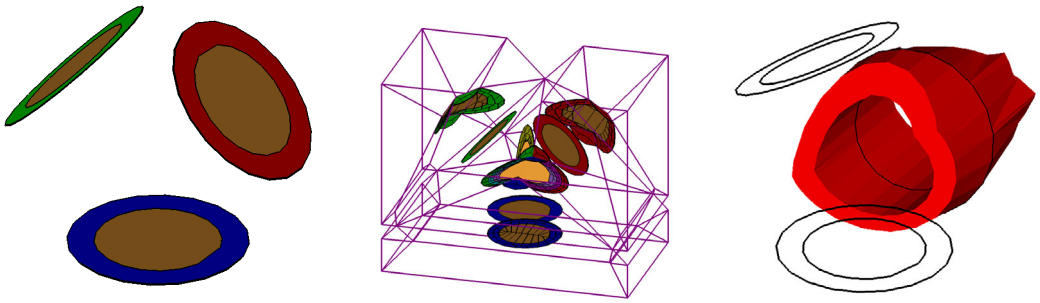


(c) An aiding blue patch (d) The overlay of the patch with the slices (e) The reconstructed mesh

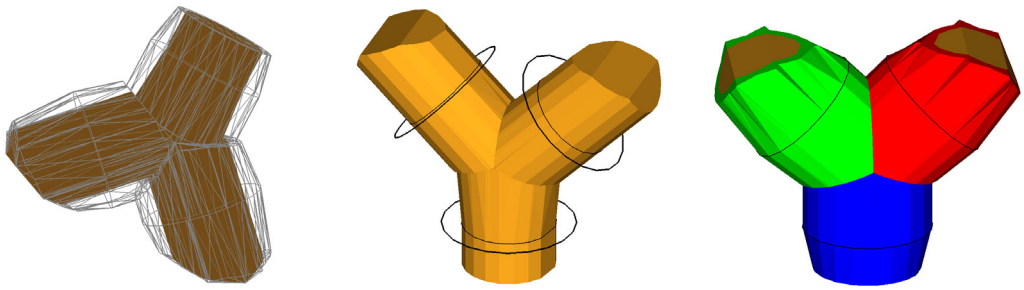
Figure 8: *Inserting a patch to aid reconstruction. The two blue stripes are expected to connect, but they do not project onto the same overlay. An aiding blue patch is put in the middle to ensure the continuity of the blue domain.*

Object	Sections	Input Vertices	Arrangement Cells	Overlays	Output Vertices	Output Triangles	Running Time(seconds)
Elephant	21	1,426	137	392	11,370	11,481	55.3
Kitten	25	982	93	234	7,543	7,862	41.03
Spheres	15	2,504	56	288	21,328	41,096	39.81
Cathead	14	549	26	167	8305	11238	29.36

Table 1: *The statistics for the experimental results.*

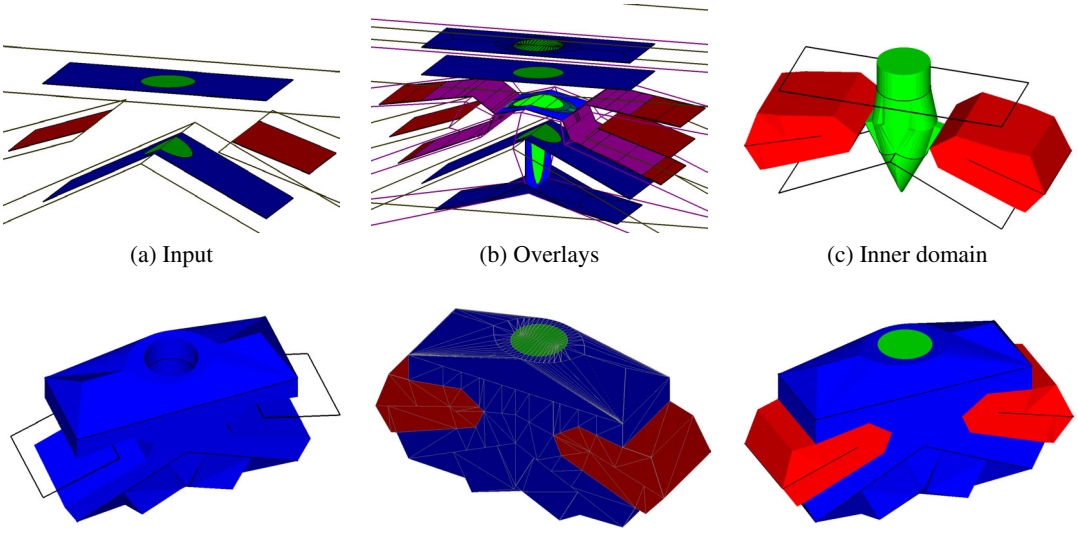


(a) Input (b) Overlays (c) One outer domain



(d) The inner domain (e) Full reconstruction

Figure 9: *Reconstruction of a "3-knuckle."*



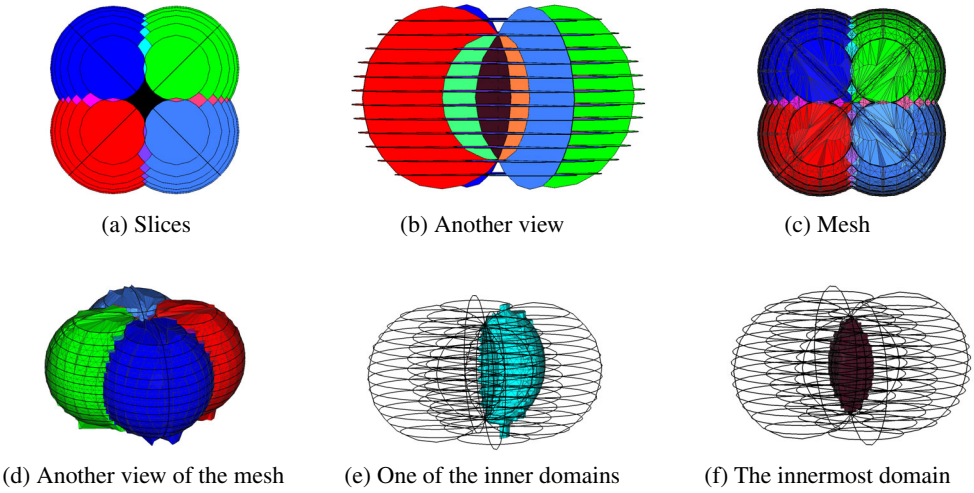
(a) Input (b) Overlays (c) Inner domain

(d) Outer domain (e) Full reconstruction

Figure 10: *Reconstruction with partial sections.*

Slices Reconstruction Smoothed

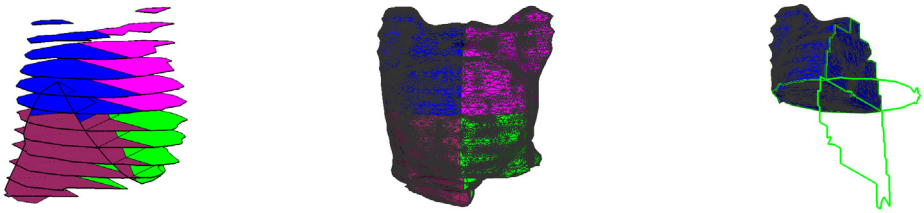
Figure 11: *Reconstruction of unicolored meshes from nonparallel sparse slices.*



(a) Slices (b) Another view (c) Mesh

(d) Another view of the mesh (e) One of the inner domains (f) The innermost domain

Figure 12: *Reconstruction of the intersection domains of four spheres.*



(a) Slices (b) The mesh (c) The blue domain

Figure 13: *Reconstruction of a mesh from partial slices. The green lines in (c) represent the non-manifold edges in the mesh.*

9. Conclusion

In this paper we present the first algorithm for reconstructing an unknown object (or objects) from cross-sections, with all the following degrees of freedom supported: (a) *nonparallel* sections; (b) *multi-colored* contours within sections; and (c) *partial* sections. Our work is built on previous work that used the straight skeleton of polyhedral objects, with some nontrivial generalizations for degenerate edges, and new special algorithmic components needed to support all the required features simultaneously. Although the mesh interpolates the cross-sections, the resulting surface can still suffer from jagged edges and sometimes poor triangulation, as most algorithms. For such cases, we applied a fairing scheme to the mesh. In addition, we cannot yet give an accurate bound on the theoretical time complexity of the algorithm since the complexity of the three-dimensional straight skeleton of a polyhedron, is still an open question. [BEGV08] suggest the upper bound $O(n^4)$ on the time needed for computing the skeleton of a nonconvex cell of complexity n (all other steps are less complicated). However, practical cases almost never admit this upper bound.

References

[AA96] O. Aichholzer, and F. Aurenhammer, Straight skeletons for general polygonal figures in the plane, *Proc. 2nd Ann. Int. Computing and Combinatorics Conf.*, Hong Kong, *Lecture Notes in Computer Science*, 1090, Springer-Verlag, 117–126, 1996.

[BCL96] C.L. Bajaj, E.J. Coyle, and K.N. Lin, Arbitrary topology shape reconstruction from planar cross sections, *Graphical Models and Image Processing*, 58 (1996), 524–543.

[BEGV08] G. Barequet, D. Eppstein, M.T. Goodrich, and A. Vaxman, Straight skeletons of three-dimensional polyhedra, *Proc. 16th Ann. European Symp. on Algorithms*, Karlsruhe, Germany, *Lecture Notes in Computer Science*, 5193, Springer-Verlag, 148–160, 2008.

[BGLS04] G. Barequet, M.T. Goodrich, A. Levi-Steiner, and D. Steiner, Contour interpolation by straight skeletons, *Graphical Models*, 66 (2004), 245–260.

[BS96] G. Barequet and M. Sharir, Piecewise-linear interpolation between polygonal slices, *Computer Vision and Image Understanding*, 63 (1996), 251–272.

[BPCC81] S. BATNITZKY, H.I. PRICE, P.N. COOK, L.T. COOK, S.J. DWYER III, Three-dimensional computer reconstruction from surface contours for head CT examinations, *J. of Computer Assisted Tomography*, 5 (1981), 60–67.

[Bo88] J.-D. BOISSONNAT, Shape reconstruction from planar cross sections, *Computer Vision, Graphics and Image Processing*, 44 (1988), 1–29.

[Bm07] J.-D. BOISSONNAT AND P. MEMARI, Shape Reconstruction from unorganized cross-sections, *Proc. Symp. on Geometry Processing*, Barcelona, Spain, 89–98, 2007.

[BTS04] A.L. BOGUSH, A. TUZIKOV, AND S. SHEYNIN, 3D object reconstruction from non-parallel cross-sections, *Proc. 17th Ann. IAPR and IEEE Int. Conf. on Pattern Recognition*, Cambridge, England, vol. 3, 542–545, 2004.

[BG92] J.D. BOISSONNAT AND B. GEIGER, Three dimensional reconstruction of complex shapes based on the Delaunay triangulation, Technical Report 1697, Inria-Sophia Antipolis, 1992.

[CD99] S.W. CHENG AND T.K. DEY, Improved construction of Delaunay based contour surfaces, *Proc. 5th ACM Symp. on Solid Modeling and Applications*, Ann Arbor, MI, 322–323, 1999.

[CGAL] Computational Geometry Algorithms Library. *http://www.cgal.org*

[CP94] Y.K. CHOI AND K.H. PARK, A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure, *The Visual Computer*, 10 (1994), 372–387.

[CS78] H.N. CHRISTIANSEN AND T.W. SEDERBERG, Conversion of complex contour line definitions into polygonal element mosaics, *Computer Graphics*, 13 (1978), 187–192.

[DP97] C.R. DANCE AND R.W. PRAGER, Delaunay reconstruction from multiaxial planar cross-sections, Technical Report CUED/F-INFENG/TR273, Cambridge Univ., England, 1997.

[EPO91] A.B. EKOULE, F.C. PEYRIN, AND C.L. ODET, A triangulation algorithm from arbitrary shaped multiple planar contours, *ACM Transactions on Graphics*, 10 (1991), 182–199.

[FO99] P. FELKEL AND Š. OBDRŽÁLEK, Improvement of Oliva's algorithm for surface reconstruction from contours, *Spring Conf. on Computer Graphics* (J. Zara, ed.), Budmerice, Slovakia, 254–263, 1999.

[FL98] J.D. FIX AND R.E. LADNER, Multiresolution banded refinement to accelerate surface reconstruction from polygons, *Proc. 14th Ann. ACM Symp. on Computational Geometry*, Minneapolis, MN, 240–248, 1998.

[FKU77] H. FUCHS, Z.M. KEDEM, AND S.P. USELTON, Optimal surface reconstruction from planar contours, *Comm. of the ACM*, 20 (1977), 693–702.

[GD82] S. GANAPATHY AND T.G. DENNEHY, A new general triangulation method for planar contours, *ACM Trans. on Computer Graphics*, 16 (1982), 69–75.

[GHH+03] M. GRANADOS, P. HACHENBERGER, S. HERT, L. KETTNER, K. MEHLHORN, AND M. SEEL, Boolean Operations on 3D Selective Nef Complexes: Data Structure, Algorithms, and Implementation. *Proc. 11th Ann. European Symp. on Algorithms* Budapest, Hungary. 654–666, 2003.

[JWC+05] T. JU, J.D. WARREN, J. CARSON, G. EICHELE, C. THALLER, W. CHIU, M. BELLO, AND I.A. KAKADIARIS, Building 3D surface networks from 2D curve networks with application to anatomical modeling, *The Visual Computer*, 21 (2005), 764–773.

[KD88] N. KEHTARNAVAZ AND R.J.P. DE FIGUEIREDO, A framework for surface reconstruction from 3D contours, *Computer Vision, Graphics, and Image Processing*, 42 (1988), 32–47.

[KSD88] N. KEHTARNAVAZ, L.R. SIMAR, AND R.J.P. DE FIGUEIREDO, A syntactic/semantic technique for surface reconstruction from cross-sectional contours, *Computer Vision, Graphics, and Image Processing*, 42 (1988), 399–409.

[Ke75] E. KEPPEL, Approximating complex surfaces by triangulation of contour lines, *IBM J. of Research and Development*, 19 (1975), 2–11.

[KCVS98] L. KOBBELT, S. CAMPAGNA, J. VORSATZ, H. P. SEIDEL, Interactive Multi-Resolution Modeling on Arbitrary Meshes, *SIGGRAPH* Orlando, FA, 105–114, USA, July 1998,

[Lp03] P. LIEPA, Filling Holes in Meshes, Proc. Symp. on Geometry Processing 2003, Aachen, Germany, 200–205, July 2003

[LBD+08] L. LIU, C. BAJAJ, J. DEASY, D.A. LOW, AND T. JU, Surface reconstruction from non-parallel curve networks, *Computer Graphics Forum*, 27 (2008), 155–163.

[MSS91] D. MEYERS, S. SKINNER, AND K. SLOAN, Surfaces from contours: The correspondence and branching problems, *ACM Trans. on Graphics*, 11 (1992), 228–258

[OPC96] J.-M. OLIVA, M. PERRIN, AND S. COQUILLART, 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized Voronoi diagram, *Computer Graphics Forum*, 15 (1996), C397–408.

[PT94] B.A. PAYNE AND A.W. TOGA, Surface reconstruction by multiaxial triangulation, *IEEE Computer Graphics Applications*, 14 (1994), 28–35.

[Sh81] M. SHANTZ, Surface definition for branching contour-defined objects, *Computer Graphics*, 15 (1981), 242–270.

[SP88] K.R. SLOAN AND J. PAINTER, Pessimal guesses may be optimal: A counterintuitive search result, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10 (1988), 949–955.

[WA86] Y.F. WANG AND J.K. AGGARWAL, Surface reconstruction and representation of 3-D scenes, *Pattern Recognition*, 19 (1986), 197–207.

[WW94] E. WELZL AND B. WOLFERS, Surface reconstruction between simple polygons via angle criteria, *J. of Symbolic Computation*, 17 (1994), 351–369.

[ZJH87] M.J. ZYDA, A.R. JONES, AND P.G. HOGAN, Surface construction from planar contours, *Computers and Graphics*, 11 (1987), 393–408.