The Visual Computer manuscript No.
(will be inserted by the editor)

Tao Ju · Joe Warren · James Carson · Gregor Eichele · Christina Thaller · Wah Chiu · Musodiq Bello · Ioannis Kakadiaris

Building 3D surface networks from 2D curve networks with application to anatomical modeling

Abstract Constructing 3D surfaces that interpolates 2D curves defined on parallel planes is a fundamental problem in computer graphics with wide applications including modeling anatomical structures. Typically the problem is simplified so that the 2D curves partition each plane into only two materials (e.g., air versus tissue). Here we consider the general problem where each plane is partitioned by a curve network into multiple materials (e.g., air, cortex, cerebellum, etc.). We present a novel method that automatically constructs a surface network from curve networks with arbitrary topology and partitioning an arbitrary number of materials. The surface network exactly interpolates the curve network on each plane and is guaranteed to be free of gaps or self-intersections. In addition, our method provides a flexible framework for user interaction so that the surface topology can be modified conveniently when necessary. As an application, we applied the method to build a high-resolution 3D model of the mouse brain from 2D anatomical boundaries defined on 350 tissue sections. The surface network accurately models the partitioning of the brain into 17 abutting anatomical regions with complex topology.

Tao Ju
Washington University, St. Louis, MO, USA
Tel.: +01 (314) 935 - 6160
E-mail: jutao@rice.edu

Joe Warren
Rice University, Houston, TX, USA
Tel.: +01 (713) 348 - 3837
E-mail: jwarren@rice.edu

James Carson, Gregor Eichele, Christina Thaller, Wah Chiu
Baylor College of Medicine, Houston, TX, USA
Tel.: +01 (713) 798 - 6989
E-mail: [james.carson][geichele][cthaller][wah]@bcm.tmc.edu

Gregor Eichele
Max Planck Institute of Experimental Endocrinology, Feodor-Lynen-Str. 7, D-30625, Hannover, Germany
Tel.: +49 (511) 5359 - 100
E-mail: gregor.eichele@mpihan.mpg.de

Musodiq Bello, Ioannis Kakadiaris
University of Houston, Houston, TX, USA
Tel.: +01 (713) 743- 1255
E-mail: [mbello][ioannisk]@uh.edu

Keywords surface network · polygonal modeling · contour interpolation

1 Introduction

A typical way of building a 3D surface model is by interpolating curves defined on parallel 2D planes. For example, in computer-aided design, a common way of creating a 3D object is by interpolating its cross-sectional curves. In geology, the surface of the terrain is often created by interpolating planar contours of the terrain at different heights. In these applications, each 2D plane contains a set of *simple closed curves* that partitions the plane into *two materials*: air and solid (or outside and inside). The resulting surface connects the curves on successive planes to form the boundary between volumes associated with the two materials. There have been tremendous amount of work in developing techniques for building such surfaces from simple planar curves, many of which have been successfully employed in CAD and geology applications (see extended review in Section 2).

In some bio-medical applications, however, we are confronted with a more demanding task. To build a 3D model of an anatomical structure, we often start from a stack of 2D tissue sections. For complex structures such as the brain, each section is annotated with various anatomical regions (e.g., cortex, cerebellum, etc.) partitioned by a network of anatomical boundaries. Figures 1 (a,b) show two annotated sections of a mouse brain where the anatomical regions are represented by colors. The goal is to connect the boundary curves on each section to form a network of surfaces in 3D, such as the one shown in Figure 1 (c), which models the partitioning of the structure into anatomical regions in space. Surface networks of this type can serve as 3D atlases of anatomical structures that often play key roles in understanding biological functions and interpreting bio-chemical data. For example, in a previous work [18] the authors have utilized a 2D atlas represented as curve networks for studying gene expression patterns over the mouse brain.

Problem statement Motivated by the biological application, we are interested in the following problem: Given a set

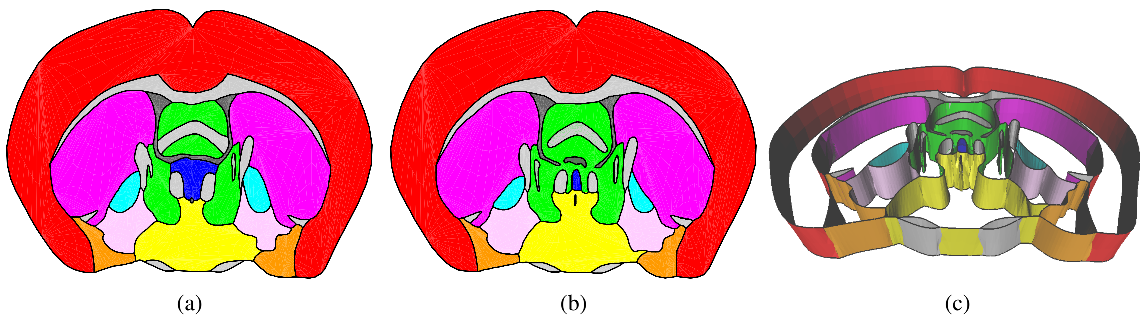


Fig. 1 Constructing a surface network from curve networks: two parallel tissue sections of the mouse brain partitioned into anatomical regions by curve networks (a,b); a surface network that interpolates the curve networks on the two sections (c).

of non-intersecting *curve networks* partitioning a series of parallel planes into *multiple materials* (e.g., anatomical regions), generate a surface network with the following properties:

Interpolation The surface network should exactly interpolate the curve networks on each plane.
Geometric correctness The surface network should partition space into disjoint volumes, each bounded by a closed mesh without self-intersections or gaps.
Topological correctness The topology of surface network should agree with the topology of the original object from which the curve networks are created.

Challenges Developing an efficient and robust method for building surface networks is a non-trivial task, especially given curve networks with complex shapes and topology (such as anatomical boundaries). In particular, direct application of existing methods for building surfaces from simple planar curves - by constructing one surface for each material partitioned by the curve network [13] - often leads to invalid geometry. For example, each of the two sections shown in Figures 2 (a,b) contains three materials representing air and two anatomical regions. Building surfaces separately for the green region and the purple region will result in self-intersecting geometry as shown in Figure 2 (c).

Moreover, given curve networks on two parallel planes, there usually exist multiple geometrically correct, yet topologically distinct surface networks. For example, Figures 2 (d,e,f) show three topologically different surface networks connecting the curve networks in Figures 2 (a,b). Since it is impossible to infer the topology of the original surface solely from the curve networks defined on the two planes, manual adjustment is often the only way to produce a valid reconstruction. Unfortunately, most existing algorithms do not provide mechanisms for user interaction (see Section 2), while the few exceptions are restricted to handle simple closed curves, instead of curve networks, on each parallel plane.

Contributions In this paper, we present a novel method for building a surface network from arbitrary curve networks defined on parallel planes. Instead of directly connecting the curves to form surfaces, which may result in invalid geometry, the key is to consider how 2D regions of the same material on successive planes are connected to form 3D volumes. We represent these volumes in space using an abstract graph structure, called the *volume graph*, which provides direct and convenient control over the topology of the resulting surface network. By adapting a multi-material contouring method [17], we are able to create a geometrically correct surface network given the topology of the volume graph. Using this method we can

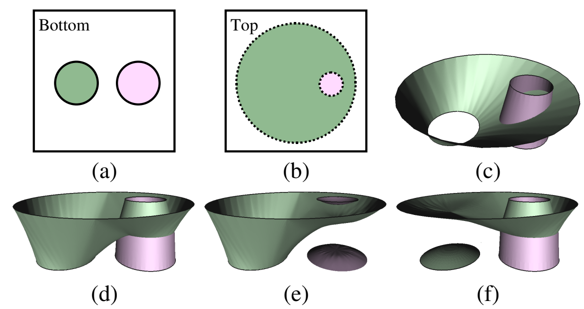


Fig. 2 Curve networks on two neighboring sections (a,b), an invalid surface network with self-intersecting geometry (c), and three geometrically correct surface networks with various topology (d,e,f).

- Automatically construct a geometrically correct surface network that interpolates arbitrary curve networks on parallel planes.
- Allow convenient user interaction to modify the topology of the automatically reconstructed surface network. Using the volume graph, the user can create a variety of surface topology without losing geometric correctness.

Some results of our method are shown in Figure 1 (c) and Figures 2 (d,e,f). As an application, we use this method to construct a high-resolution model of the adult mouse brain. The model is created automatically from 350 brain sections followed by user-guided topology adjustment, and the resulting surface network accurately models the partitioning of the brain into 17 anatomical regions with complex shapes and topology.

2 Previous work

Building surfaces from curves on parallel planes has been studied extensively in the past three decades, and numerous methods have been proposed. In many fields the problem is also known as *contour interpolation* or *contour tiling*, where the curves on each plane are referred to as contour lines [19]. Here we attempt only to provide a brief review of some of these methods, and we refer interested readers to excellent surveys by Hagen [14] and by Sloan and Painter [24].

With few exceptions [26], previous methods are designed to build a surface from simple contour lines that partition each plane into only two materials (e.g., air and solid). Early approaches attempt to find a triangular tiling that connects the contour lines on the neighboring planes while optimizing a quality measure, such as the minimum surface area criteria in the work of Fuchs *et al.* [12]. While Fuchs' and later approaches [9,21] produce fairly good looking surfaces connecting simple contour lines, the techniques may generate surfaces with either self-intersections or gaps between planes containing multiple and nested contours.

Recent work on surface reconstruction from simple planar curves focuses on robust handling of curves with arbitrary topology and geometry. Boissonnat presents a method based on Delaunay triangulations [6], which is refined by Geiger [13]. Cheng *et al.* [8] improve Boissonnat's method further to generate surfaces without having to compute 3D Delaunay triangulations. Several researchers have used implicit functions to interpolate between planar curves, such as the method of Herman *et al.* [15] and Csebfalvi *et al.* [10]. The variational approach of Turk *et al.* [25] combines the two steps of building and interpolating implicit functions. Generating smooth surfaces by solving Partial Differential Equations (PDE) has also been considered by Bloor *et al.*[5] and later by Chai *et al.* [7], who further studied smooth connections of neighboring surfaces sharing a common curve. Finally, a group of researchers ([2–4,20,23]) have developed methods based on computing orthogonal projections of two neighboring planes in order to construct surfaces with correct geometry and reasonable topology. In particular, the methods of Oliva *et al.* [23] and Barequet *et al.* [3] compute the areas of differences on the projected plane and triangulate these areas using Voronoi diagrams [23] or Straight Skeletons [3].

To the best of our knowledge, the only method for handling curve networks that partition each underlying plane into multiple materials is the recent work by Weinstein [26]. Using a volumetric approach, Weinstein first voxelizes each plane onto a uniform grid, where each grid point is associated with a material type. Next, the surfaces between volumes of different materials are generated using contouring on the voxel grid. Weinstein's approach produces geometrically correct surface network for curve networks of arbitrary topology. However, due to the use of regular voxels, the surface network produced by Weinstein's approach only approximates rather than interpolates the curve network on each plane.

Most existing algorithms (including the work by Weinstein [26]) are designed to completely automate surface reconstruction, leaving little room for the user to adjust the topology of the resulting surface network. The few exceptions include the method of Christiansen and Sederberg [9], where user interaction is required to guide the triangulation in cases of excessive ambiguity. Software packages, such as SURFdriver [22], allow a limited degree of user interaction during surface reconstruction, such as capping and connecting regions. However, these mechanisms for user interaction are restricted to simple closed curves on each plane, and can be hard to extend to curve networks due to the increased complexity of surface topology.

3 Algorithm overview

Like most other methods for building surfaces from curves on parallel planes, our method computes a layer of surface network between every two neighboring planes and concatenates successive layers to form a complete model. Given curve networks defined on two neighboring planes (Figures 3 (a,b)), our method proceeds in three steps:

1. **Projection:** Project the curve networks from each plane orthogonally onto a common plane. A *wedge* is defined as a cylindrical space between the two planes projecting onto a partitioned region on the common plane (Figure 3 (c)).
2. **Topology creation:** Create a *volume graph* that describes how each wedge is decomposed orthogonally into *layers* of materials, and how layers of the same material between neighboring wedges are connected to form volumes (Figure 3 (d)).
3. **Polygonalization:** Construct the surface network by polygonalizing the interfaces between neighboring volumes in the volume graph (Figure 3 (f)).

The key step in our method is creating the volume graph, which describes how the space between the two planes is partitioned into volumes of various materials. Representing each layer as a node, the volume graph expresses the partitioning in an intuitive manner: connected nodes of the *same* material in the graph represent a continuous volume, while connected nodes of *different* materials represent an interface between neighboring volumes (Figure 3 (d)). As a result, the surface network corresponds to only edges on the graph exhibiting a material change.

Given a set of planar curve networks, the three steps are performed automatically by creating and using an initial volume graph (such as Figure 3 (d)). However, one can iterate step 2 and 3 by modifying the volume graph (manually) and regenerating the surface geometry (automatically). For example, Figure 3 (e) shows a modified volume graph from the default graph in (d), and the resulting surface is shown in (g). Note that the polygonalization step guarantees to create a geometrically correct surface given any valid volume graph.

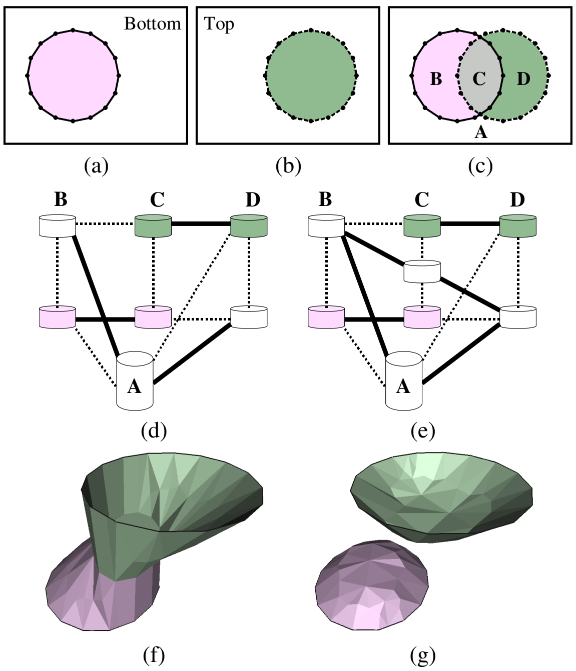


Fig. 3 Surface generation: curve networks on two neighboring planes (a,b), orthogonal projections of the curves on a common plane with the partitioned wedges (c), the initial volume graph (d) and the polygonalized surface network (f), a modified volume graph (e) and the resulting surface network (g). The surfaces correspond to the dotted edges in the volume graph.

Next we shall discuss the three steps of our method in detail, followed by an application of the method to build a surface network representing the mouse brain for bio-medical studies.

4 Projection

Given two neighboring planes, we compute a decomposition of the space between the planes into *wedges* using the orthogonal projection of the curve networks. To do so, we first compute intersections of the projected curves from two planes onto a common plane. The projections of these intersection points are added back into the curve networks on each plane. To maintain topological connectivity between surface networks generated between successive planes, it is important that the curves on each plane are augmented with projections of the intersection points with curves from its two neighboring planes. In this way, the polygons generated on the two sides of the plane will share common vertices.

Second, the partitioned regions (such as the labeled regions in Figure 3 (c)) are identified on the common plane by tracing closed boundary cycles in the projected curve network (a region may be bounded by multiple boundary cycles, which can be detected using a scan-line based algorithm [11]). Each partitioned region then corresponds to a wedge of space bounded between the two planes that projects onto the region.

5 Topology creation

The key idea behind our method is to consider how the space between two planes is partitioned into volumes of various materials. Since this space is already divided into wedges, we further decompose each wedge into *layers* stacked in the orthogonal direction from the bottom plane to the top plane. Each layer is an atomic element of the space filled with a single material, and a continuous volume is formed by layers of a same material between neighboring wedges. We create a graph (called the *volume graph*) to describe this formation of volumes:

- A **node** in the graph represents a layer of a wedge and is associated with the layer's material.
- An **edge** in the graph connects neighboring layers within a wedge or between neighboring wedges.

The volume graph in Figure 3 (d) is automatically created from the two planes in (a,b), where the top (bottom) nodes of each wedge are associated with the corresponding materials on the top (bottom) plane and are connected by edges between neighboring wedges. Note that the white nodes are created to represent the "outside" material. To aid visualization, the nodes representing layers of a same wedge are organized along a vertical line. Observe that the structure of the graph defines the topology of the volumes as well as that of the resulting surface network:

- Each connected component of nodes of the *same* material (connected by solid lines in Figure 3 (d)) represents a continuous volume.
- Each edge between nodes of *different* materials (shown as dotted lines in Figure 3 (d)) represents a boundary surface between two neighboring volumes.

For example, the volume graph in Figure 3 (d) contains three connected components of nodes associated with materials white, purple and green, corresponding to the three volumes partitioned by the resulting surface network in Figure 3 (f). On the other hand, the edge connecting the two nodes of wedge **C** in Figure 3 (d) corresponds to the boundary between the green and purple volumes in the middle of Figure 3 (f).

5.1 Graph variation

The initial volume graph created automatically can be altered to yield various surface topology in several ways:

Adding nodes and edges Inserting new nodes between the top and bottom nodes of each wedge will effectively build intermediate layers of structures. Since the corresponding volumes on the resulting surface network are inherently disjoint, intersecting edges are prohibited in the volume graph. That is, let the nodes of each wedge be indexed from bottom to top with increasing indices, if two edges connect the ith and jth node of one wedge to the kth and lth node of a neighboring wedge, we require $(i-j)*(k-l) \geq 0$. In addition, to avoid introducing dangling structures, we further require

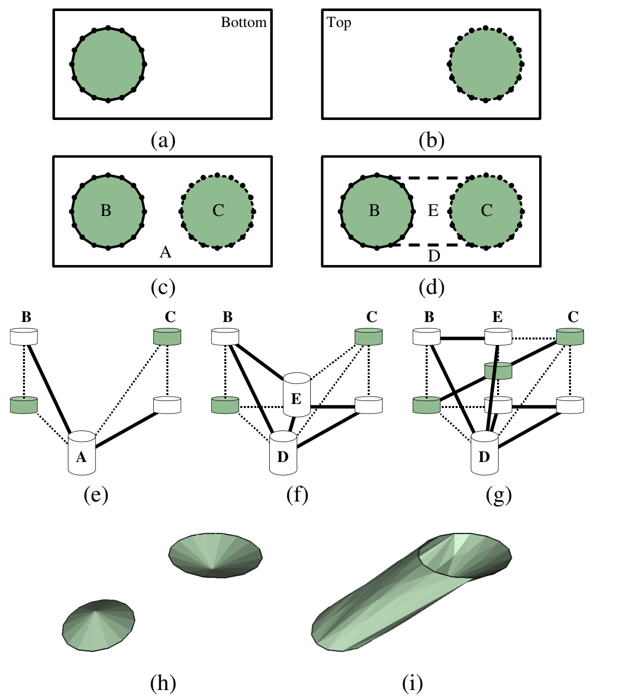


Fig. 4 Topology modification through wedge splitting: two planes containing non-overlapping regions of the same material (a,b), their orthogonal projections before (c) and after (d) wedge splitting, the initial volume graph (e) and after wedge splitting (f) and further modification (g), the resulting surface network before (h) and after (i) modification.

that each newly inserted node must be connected by edges to existing nodes of that material on neighboring wedges.

For example, we can insert a white node to wedge **C** in the graph of Figure 3 (d) and connect the new node to the two white nodes in neighboring wedges **B** and **D**, as shown in Figure 3 (e). While the purple and green volumes share a common boundary in the original graph (d), they are separated by a continuous volume of white material in the modified graph (e), as reflected in the resulting surface network in (g).

Splitting wedges When adding nodes and edges is not sufficient to create surface network with a desired topology, we may split an existing wedge into smaller wedges. Accordingly, the volume graph is *populated* to include nodes and edges representing the split wedges. This is often useful, for example, in creating tubular structures that connect distant features on the two planes.

As an example, the curves shown in Figures 4 (a,b) result in two pieces of surfaces, shown in (h), since the projections of the two curves, shown in (c), do not overlap. To create a single tube that connects the two curves, we divide wedge **A** in (c) into two smaller wedges, **D** and **E**, shown in (d), and create a populated graph (f) by duplicating the nodes and edges of wedge **A** in the original graph (e) for the two new wedges. Then, we can modify the populated graph by adding nodes and edges, as shown in (g), to form a continuous volume with a tubular surface, as shown in (i).

6 Polygonalization

Once the topology of the surface network is defined by the volume graph, our final task is to create the actual geometry. Based on our observation in the previous section, we shall construct polygons that correspond to edges on the volume graph connecting nodes of different materials. However, the abstract structure of the volume graph does not permit direct creation of polygonal surfaces. Our strategy is to turn the volume graph into a concrete volumetric grid from which polygons (e.g., triangles and quads) can be extracted using contouring methods.

6.1 Planar triangulation

We consider the orthogonal projections of the wedges on the common plane in Figure 3 (c). By applying 2D triangulation for each region on the plane, as shown in Figure 5 (a), we effectively split each 3D wedge into triangular prisms. Using the wedge splitting technique discussed in the previous section, the initial and the modified volume graph in Figures 3 (d,e) are populated to form *volume grids*, a portion of which is shown in Figures 5 (b,d) corresponding to the shaded prisms in Figure 5 (a). Note that the nodes and edges of each prism are inherited from the original wedges in the volume graph to which the prisms belong.

In our implementation, we used the Straight Skeleton triangulation method proposed by Aichholzer and Aurenhammer [1] and adopted by Barequet *et al.* [3] for contour interpolation. Note that we do not need to triangulate wedge **A** in Figure 5 (a) since it does not contain an edge on the volume graph and hence will not contribute to the surface network.

6.2 Multi-material contouring

To extract polygons corresponding to edges on the volume grid exhibiting a material change, we consider the multi-material Dual Contouring (DC) method of Ju *et al.* [17] for voxel grids. Given a voxel grid consisting of cubic cells where every grid point is associated with a material, the DC method constructs a surface network in two steps:

1. Create one vertex for each grid cell.
2. Create one polygon for each grid edge exhibiting a material change by connecting the vertices of the cells sharing that edge.

The DC method guarantees to partition the grid into disjoint volumes of different materials in the presence of an arbitrary number of materials on the grid, and can be easily extended to an adaptive voxel grid, such as an octree.

To adapt the DC method to our volume grid that has no voxelized structure, we need to first define three more topology elements on the volume grid: columns, faces and cells. First, each portion of grids shown in Figures 5 (b,d) is called a *column*, which consists of nodes for all prisms sharing a common edge. Each column is centered at a vertex on the common plane shared by the triangular projections of the prisms. A column is surrounded by grid *faces*, each being a

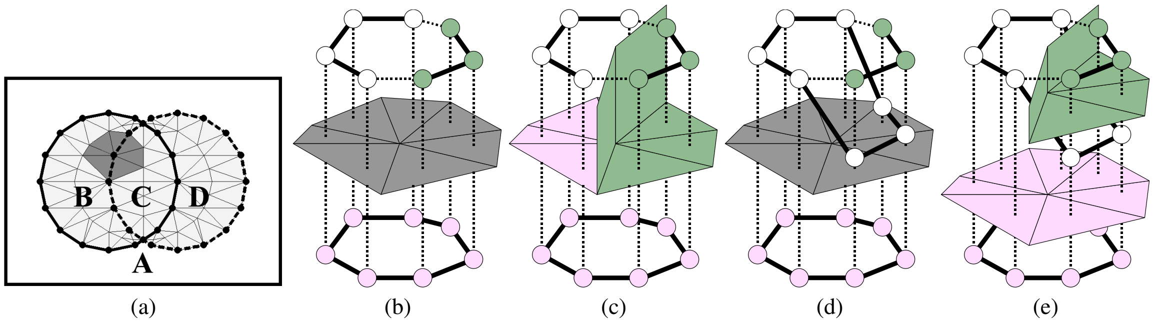


Fig. 5 Surface polygonalization: A triangulation of the projected regions on the common plane (a), portions of the volume grids (b,d) corresponding to the shared triangles in (a) created from the initial and modified volume graph from Figure 3, and the surface generated on the grids (c,e).

closed cycle of grid edges between two neighboring prisms. For example, the columns in Figures 5 (b,d) are surrounded by 7 and 11 faces respectively. A column is also partitioned vertically into *cells* by rings of edges connecting nodes of the same material. For example, the columns in Figures 5 (b,d) contain one and two cells respectively. In addition, the cells in a same column are indexed vertically from bottom to top. To ensure unique indexing, we require the ith cell and the $(i+1)$st cell share a ring of edges that completely surrounds the column, such as the ring of edges connecting white nodes between the two cells in the column of Figure 5 (d).

Given the definitions of the topology elements, we are now in place to describe the surface generation algorithm. Assuming the top and bottom planes on which the curve networks lie are at $z = 0$ and $z = 1$ and the common plane onto which the curve networks projected is at $z = 0.5$, the algorithm proceeds in two simple steps,

1. **Create one vertex for each cell on the volume grid.** The ith cell ($i = 1, \ldots, n$) on the grid column centered at the vertex $\{x, y, 0.5\}$ on the common plane have coordinates $\{x, y, (2i-1)/2n\}$. In addition, we create a pair of *pseudo vertices* at $\{x, y, 0\}$ and $\{x, y, 1\}$ for each column.
2. **Create one polygon for each edge on the volume grid that exhibits a material change.** The polygon connects the vertices of the grid cells sharing that edge. For an edge at the top (bottom) of a grid column, a quadrilateral is formed by connecting the vertices of the two cells sharing the edge and the top (bottom) pseudo vertices of the two columns sharing the edge.

Figures 5 (c,e) show the surfaces generated on the grid columns in (b,d) using this method. Observe that since vertices within a same grid column differ only in their Z coordinates, the polygons on the 3D surface project exactly onto the triangles in the 2D triangulation shown in Figure 5 (a). In this sense, the 3D surface is created by *lifting* the 2D triangles in space (a similar mechanism was originally used by Barequet [3] yet restricted to only simple closed curves on each plane). Next we discuss the two important properties of the surface network that we desire to have as outlined Section 1:

Interpolating curve networks To show that the surface network created by our method interpolates the curve networks on the top and bottom plane, we first note that the vertices of the curve networks (including the intersection points augmented during projection) is a subset of the pseudo vertices. Our surface network is connected to the top (bottom) plane only at edges of quadrilaterals connecting pseudo vertices lying on the boundary of different materials, which are the segments of the curve networks on that plane.

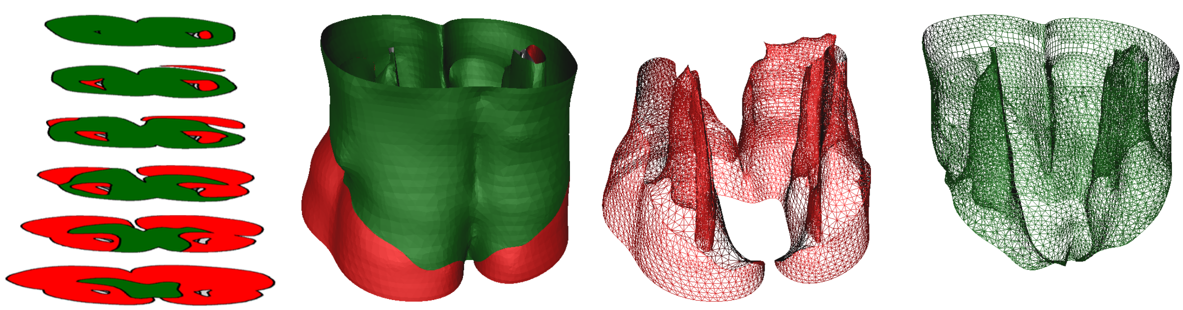
Geometric correctness Observe from Figures 5 (c,e) that each grid edge exhibiting a material change corresponds to a polygon on the surface, while each grid face exhibiting a material change corresponds to an edge on the surface. The duality results in a gap-free surface network that partitions the space into disjoint volumes corresponding to connected components of grid points of a same material.

To show that the polygonal surface is also free of self-intersections, we first observe that the only possible self-intersection is between two triangles on the surface projecting onto the same triangle on the common plane. Yet this will not take place: suppose such two triangles are dual to the ith and jth edge (indexed from bottom to top) of a same prism on the volume grid with $i < j$. In every column that contains the two edges, the grid cells containing edge i will always have a lower or same index as the cells containing edge j. As a result, a vertex in the triangle dual to edge i always has a lower or same Z coordinate with the vertex (with the same X, Y coordinates) in the triangle dual to edge j.

6.3 Surface smoothing

The algorithm in the previous section produces vertices with uniform spacing in the Z direction, which may result in a jagged appearance where polygons meet at right angles as seen in Figures 5 (c,e). To achieve a smoother-looking surface while maintaining geometric correctness, we apply a simple Laplacian smoothing operator *along the Z direction* for every non-pseudo vertex v:

$$v_z^* = \frac{v_z}{2} + \frac{\sum_{w \in N(v)} w_z}{2\|N(v)\|}$$



(a) (b) (c) (d)

Fig. 6 First 20 cross-sections of the mouse brain (only showing a few) (a), the reconstructed surface network (b), the surface bounding Cortex (c) and Olfactory bulb (d) (the distances between successive cross-sections have been exaggerated for better visualization).

where v_z denotes the Z coordinate of vertex v and $N(v)$ denotes the set of edge-adjacent neighbors v. Note that smoothing the Z-coordinates of vertices will not cause self-intersection on the smoothed surface so long as the monotonicity of cell ordering is maintained: the Z coordinate of the vertex of the ith cell in each grid column is strictly larger than that of the $(i-1)$st cell in the same column.

We can further increase the surface smoothness by reducing the number of quadrilaterals that connect the surface network to the curve networks at pseudo vertices. These quadrilaterals always lie perpendicular to the input planes and their orientations are not affected by the Laplacian smoothing in the Z direction. An effective strategy that we found is to move the first (last) vertex in each grid column to lie on the bottom (top) plane whenever possible, so that the majority of the quadrilaterals will degenerate to a line and will not contribute to the surface geometry. The final surface networks corresponding to the default and modified volume graph in Figures 3 (d,e) after quad removing and Laplacian smoothing are shown in Figures 3 (f,g).

7 Results

Here we present the result of applying the proposed method to our motivating application for building a high-resolution 3D model of the mouse brain. A stack of 350 tissue sections of the mouse brain was collected in Dr. Eichele's lab at Baylor College of Medicine, each annotated with a subset of 17 major anatomical regions [1]. Figures 1 (a,b) show two neighboring sections in this stack.

The input curve networks contain 200404 vertices on 350 sections, and these curve networks define 18 materials (17 anatomical regions plus one empty space) in total. The output surface contains 1132731 polygons and 590902 vertices. The algorithm is implemented in *Mathematica* (version 4.0), a numerical and symbolic computation environment [27], for robust handling of numerical calculations using arbitrary precision numbers. The total reconstruction took 344 minutes and 45 seconds, averaging 59 seconds between two neighboring sections (see Section 8 for more discussion on performance).

Figure 6 shows the result of automatically reconstructing the first 20 sections of the mouse brain. Observe in (a) that the interaction between Cortex (colored red) and Olfactory Bulb (colored green) changes dramatically from the top to bottom. The reconstructed surface network shown in (b,c,d) accurately models these complex anatomical regions as well as their adjacency relations.

Figures 7 (a,b) show the side and bottom views of the surface network automatically constructed using our method for the entire stack. The surface network partitions space into anatomical regions whose cross-sections coincide with the annotation on each of the 2D sections. To better illustrate this partitioning, we isolate a complex anatomical region, Fiber Tracks, in Figure 7 (c). As an advantage of the surface network representation, we can color the surface of the Fiber Tracks using its abutting anatomical regions. The coloring reveals the 3D adjacency relation between Fiber tracks and its neighboring anatomy. We believe that this high-resolution model represented as surface networks will be helpful for anatomists and brain researchers.

Due to the small distance between adjacent 2D sections in the mouse brain (i.e. 25μm), the automatically constructed surface network possesses the correct topology in most places. However, the reconstructed surface fails to produce desired topology occasionally when there is a large migration of anatomical features between successive sections. For example, Figure 8 (a) shows a tubular portion of Ventricles that is broken into separate pieces after automatic reconstruction due to a large displacement of corresponding anatomical regions on successive sections (again, the surface is colored by its abutting anatomical structures). The surface was corrected using the wedge splitting technique discussed in Section 5.1, and the regenerated surface, shown in Figure 8 (b), models a continuous tubular structure. In another example, Figure 9 (a) shows a sheet-like portion of Ventricles containing a "hole" (like one in a torus) after automatic reconstruction. A continuous sheet is recovered in Figure 9 (b) as a result of addition of nodes and edges on the volume graph

[1] The annotated regions are: Cortex, Cerebellum, Striatum, Basal Forebrain, Amygdala, Hippocampus, Hypothalamus, Thalamus, Olfactory Bulb, Midbrain, Pons, Medulla, Ventral Striatum, Globus Pallidus, Septum, Fibers, and Ventricles.

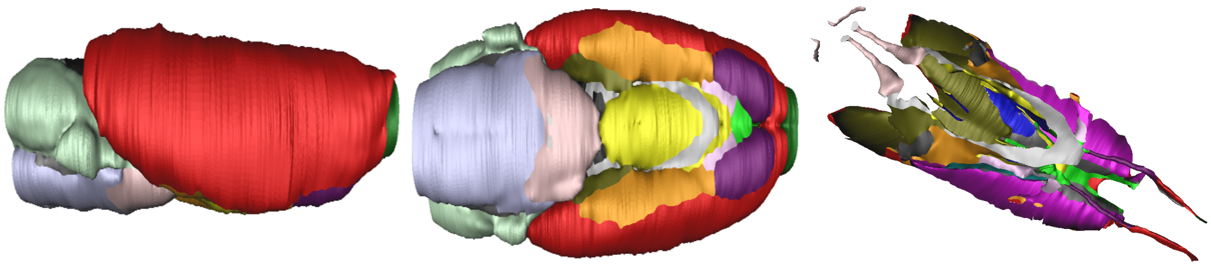

(a) (b) (c)

Fig. 7 Surface reconstruction of the entire mouse brain viewed from side (a) and bottom (b), and the reconstructed Fiber Tracks (c) colored using its neighboring anatomical regions.

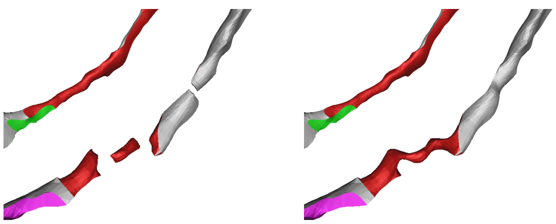

(a) (b)

Fig. 8 Broken pieces of Ventricle resulting from automatic construction (a) and after manual adjustment (b).

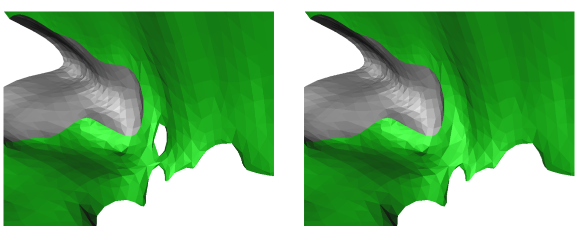

(a) (b)

Fig. 9 Disrupted sheet of Ventricle resulting from automatic construction (a) and after manual adjustment (b).

as discussed in Section 5.1. We made 11 such topological changes on the mouse brain model, and the amount of human intervention time is less than one hour in total.

8 Discussion

In this paper we present a novel technique for building surface networks from planar curve networks that often arise in bio-medical modeling applications. While previous methods often result in invalid surface geometry or fail to provide interaction mechanisms, the presented method is capable of creating a geometrically correct surface network automatically from arbitrary curve networks on parallel planes, and providing easy, flexible means for user-guided modifications to produce desirable surface topology. As an example, we show how the method is applied to build a high resolution model of the mouse brain from annotated 2D tissue sections.

One drawback of the method is that the users have to manually check and correct invalid topologies of the reconstructed surface. A major direction for future research is therefore to automate this process by enforcing certain rules on the structure of the volume graph. A simple example would be to always connect a particular material (e.g. a blood vessel) on successive planes. More complicated examples includes tracking feature points (e.g. points where three or more materials meet) on successive curve networks.

While the method is implemented in *Mathematica* for numerical robustness, we plan to port the program to C/C++ for a significant improvement in efficiency and for ease of distribution. To solve the potential problem of numerical instability, we plan to adopt a technique similar to that in [16] by discretizing the vertices in the input curve networks on a fine integer grid and to use Separating Axis theorem for intersection tests.

Finally, while most current meshing processing techniques are designed exclusively for simple closed surfaces, we plan to investigate the extension of these methods onto a surface network. For example, as shown in Figure 7, the surface network exhibits ripples due to the distortions in the input tissue sections induced by the physical sectioning process. Moreover, the size of the mesh is too large for many applications. We shall consider as the next step adapting existing mesh fairing and simplification methods from simple surfaces to surface networks. In addition, tetrahedralization of the surface network is also a potential topic of research.

Acknowledgements This work is supported in part by a training fellowship (NLM Grant No. 5T15LM07093) from the W.M. Keck Foundation to the Gulf Coast Consortia through the Keck Center for Computational and Structural Biology. It is also supported by the Burroughs Wellcome Fund, NIH grants (P41RR02250, HD24064 and DK56338), and NSF grants (ITR-0205671 and EIA-0325004). The work of Gregor Eichele is also supported by the Max-Planck-Society.

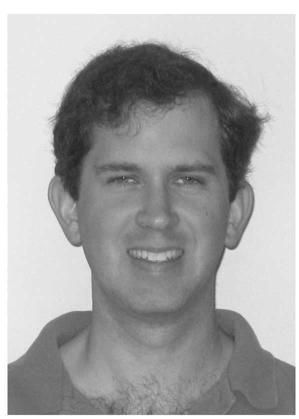
References

1. Aichholzer, O., Aurenhammer, F.: Straight skeletons for general polygonal figures in the plane. In: Computing and Combinatorics, pp. 117–126 (1996). URL citeseer.ist.psu.edu/aichholzer96straight.html
2. Bajaj, C.L., Coyle, E.J., Lin, K.N.: Arbitrary topology shape reconstruction from planar cross sections. Graph. Models Image Process. **58**(6), 524–543 (1996). DOI http://dx.doi.org/10.1006/gmip.1996.0044

3. Barequet, G., Goodrich, M.T., Levi-Steiner, A., Steiner, D.: Straight-skeleton based contour interpolation. Graphical Models **65**, 323–350 (2004)
4. Barequet, G., Sharir, M.: Piecewise-linear interpolation between polygonal slices. Computer Vision and Image Understanding **63**, 251–272 (1996)
5. Bloor, M., Wilson, M.: Local control of surfaces generated using partial differential equations. Computer Graphics **18**(3), 161–169 (1994)
6. Boissonnat, J.D.: Shape reconstruction from planar cross sections. Comput. Vision Graph. Image Process. **44**(1), 1–29 (1988)
7. Chai, J., Miyoshi, T., Nakamae, E.: Contour interpolation and surface reconstruction of smooth terrain models. In: VIS '98: Proceedings of the conference on Visualization '98, pp. 27–33. IEEE Computer Society Press (1998)
8. Cheng, S.W., Dey, T.K.: Improved constructions of delaunay based contour surfaces. In: SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications, pp. 322–323. ACM Press (1999). DOI http://doi.acm.org/10.1145/304012.304061
9. Christiansen, H.N., Sederberg, T.W.: Conversion of complex contour line definitions into polygonal element mosaics. Computer Graphics **13**, 187–192 (1978)
10. Csébfalvi, B., Neumann, L., Kanitsar, A., Gröller, E.: Smooth shape-based interpolation using the conjugate gradient method. In: Proceedings of Vision, Modeling, and Visualization, pp. 123–130 (2002)
11. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics (2nd Edition). Addison-Wesley Pubication Company (1990)
12. Fuchs, H., Kedem, Z.M., Uselton, S.P.: Optimal surface reconstruction from planar contours. Commun. ACM **20**(10), 693–702 (1977). DOI http://doi.acm.org/10.1145/359842.359846
13. Geiger, B.: Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. Tech. Rep. RR-2105. URL citeseer.ist.psu.edu/geiger93threedimensional.html
14. Hagen, H.: Focus on Scientific Visualization. Springer Verlag (1993)
15. Herman, G.T., Zheng, J., Bucholtz, C.A.: Shape-based interpolation. IEEE Comput. Graph. Appl. **12**(3), 69–79 (1992). DOI http://dx.doi.org/10.1109/38.135915
16. Ju, T.: Robust repair of polygonal models. ACM Trans. Graph. **23**(3), 888–895 (2004). DOI http://doi.acm.org/10.1145/1015706.1015815
17. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of hermite data. In: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 339–346. ACM Press (2002). DOI http://doi.acm.org/10.1145/566570.566586
18. Ju, T., Warren, J., Eichele, G., Thaller, C., Chiu, W., Carson, J.: A geometric database for gene expression data. In: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 166–176. Eurographics Association (2003)
19. Keppel, E.: Approximating complex surfaces by triangulation of contour lines. IBM JOURNAL OF RESEARCH AND DEVELOPMENT **XIX**, 2–11 (1975)
20. Klein, R., Schilling, A., Straßer, W.: Reconstruction and simplification of surfaces from contours. Graphical models **62**(6), 429–443 (2000). URL citeseer.ist.psu.edu/klein00reconstruction.html
21. Meyers, D., Skinner, S., Sloan, K.: Surfaces from contours. ACM Trans. Graph. **11**(3), 228–258 (1992). DOI http://doi.acm.org/10.1145/130881.131213
22. Moody, D., Lozanoff, S.: Surfdriver: A practical computer program for generating three-dimensional models of anatomical structures
23. 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), 397–408 (1996)
24. Sloan, K.R., Painter, J.: From contours to surfaces: testbed and initial results. In: CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, pp. 115–120. ACM Press (1987). DOI http://doi.acm.org/10.1145/29933.30869
25. Turk, G., O'Brien, J.F.: Shape transformation using variational implicit functions. In: SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 335–342. ACM Press/Addison-Wesley Publishing Co. (1999). DOI http://doi.acm.org/10.1145/311535.311580
26. Weinstein, D.: Scanline surfacing: Building separating surfaces from planar contours. In: VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000). IEEE Computer Society (2000)
27. Wolfram, S.: The Mathematica Book. Wolfram Media/Cambridge University Press (1993)

Tao Ju Tao Ju graduated from Tsinghua University in 2000 with a BA degree in English and a BS degree in Computer Science. He received his Ph.D degree in Computer Science from Rice University in 2005. Tao is currently an assistant professor in the Department of Computer Science and Engineering at Washington University in St. Louis. His research interests are in the areas of mesh processing, visualization, geometric modeling, and their applications in biology and medicine.

Joe Warren Joe Warren, a Professor of Computer Science at Rice University, is one of the world's leading experts on subdivision. He has published numerous papers of this topic and its applications to computer graphics. These publications have appeared in such forums as SIGGRAPH, Transactions on Graphics, Computer-Aided Geometric Design and The Visual Computer. He has also organized and participated in a number of international workshops, short courses and minisymposia on the theory and practice of subdivision. Professor Warren's related areas of expertise include computer graphics, geometric modeling and visualization.



James Carson James Carson received his Bachelor and Master degrees in Electrical Engineering from Rice University, and his Ph.D. from the Structural and Computational Biology & Molecular Biophysics program at Baylor College of Medicine. His doctoral dissertation was titled "Quantitative Annotation and Analysis of Gene Expression Patterns with an Atlas of the Mouse Brain". James is currently a W. M. Keck Center for Computational & Structural Biology Postdoctoral Fellow.