Beta-Connection: Generating a Family of Models from Planar Cross Sections

LUIS GUSTAVO NONATO, ALEX JESUS CUADROS-VARGAS, ROSANE MINGHIM, and
MARIA CRISTINA F. DE OLIVEIRA
Universidade de São Paulo

Despite the significant evolution of techniques for 3D-reconstruction from planar cross sections, establishing the correspondence of regions in adjacent slices remains an important issue. In this article, we propose a novel approach for solving the correspondence problem in a flexible manner. We show that from the 3D Delaunay triangulation, it is possible to derive a distance measure among regions lying in adjacent slices. Such distance is used to define a positive integer parameter, called β, responsible for establishing the connections. Varying β thus allows the construction of different models from a given set of cross-sectional regions: small values of β causes closer regions to be connected into a single component, and as β increases, more distant regions are connected together. The algorithm, named β-connection, is described, and examples are provided that illustrate its applicability in solid modeling and model reconstruction from real data. The underlying reconstruction method is effective, which jointly with the β-connection correspondence strategy, improve the usability of volumetric reconstruction techniques considerably.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling

General Terms: Algorithms

Additional Key Words and Phrases: Cross sections, Delaunay triangulation, 3D reconstruction

1. INTRODUCTION

Over the years, many algorithms have been developed to reconstruct three-dimensional models from a sequence of parallel planar cross sections of an object. Many such algorithms use the boundaries of regions of interest to obtain three-dimensional models. Reconstruction from contour boundaries creates models that may be ensured to match very precisely the regions of interest and their boundaries. This is a fundamental requirement in many applications involving numerical simulations where a rigorous definition of the simulation domain is crucial. This is the case of procedure simulations in medicine and other health and earth sciences, for example, a blood-flow simulation in human arteries or a simulation of water infiltration in soil. Detecting the regions of interest automatically may not be effective in this kind of application, making a segmentation step in each data slice necessary. Once the segmentation has been carried out, it is natural to make use of reconstruction from contours to generate the models.

This research was performed with the support of FAPESP—the State of São Paulo Research Funding Agency (Grant No. 03/02815-0), and CNPq, the Brazilian National Research Council (Grants Nos. 521931/97-5 and 300531/99-0).
Authors' address: Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - USP - São Carlos, Av. Trabalhador São-Carlense 400, 13.560-970 - Brazil; email: {gnonato,alexj,rminghim,cristina}@icmc.usp.br.

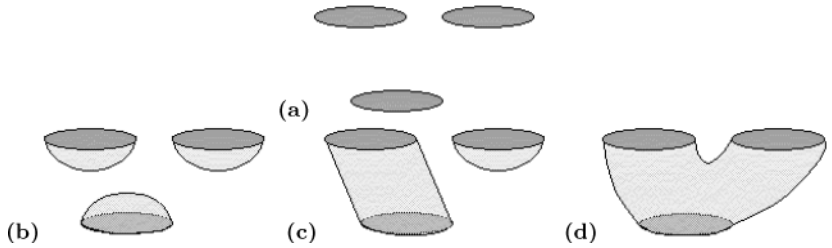

Fig. 1. Objects with identical planar cross sections.

In order to be effective, algorithms that reconstruct three-dimensional models from contours must handle three intrinsic problems, namely, correspondence, branching, and tiling. The correspondence problem arises when it is necessary to decide which regions in adjacent slices must be connected. Branching is related to the modeling of saddle points that may appear in the models. Tiling deals with constructing a mesh that connects the planar regions in adjacent sections.

Although the reconstruction algorithms described in the literature present a range of different strategies to solve the branching and tiling problems, very little has been done to handle the correspondence problem in a flexible manner. Almost every reconstruction technique either supposes that the user previously defines the correspondence or assumes the existence of a proximity criterion to automatically decide how to connect regions contained in adjacent planar sections.

Flexibility in the choice of the correspondence among regions is very important as the information in the cross sections is not always sufficient to guarantee the correct connections, that is, the ones that really correspond to the actual object being modeled. This fact is illustrated in Figure 1, where the slices in Figure 1(a) could be obtained from any one of the objects shown in Figures 1(b), 1(c), or 1(d). A less rigid approach for establishing the correspondence is highly desirable in applications such as reconstruction of arterial structures and solid modeling. In many situations arising in these and other domains, one does not know for certain which solution approximates best the original model, and the user should be able to consider and investigate multiple alternatives.

In this work, we present a reconstruction method with a new flexible approach to tackle the correspondence problem. The strategy relies on the construction of a three-dimensional Delaunay triangulation from the regions contained in the planar cross sections. We show, as an important related contribution, that the 3D Delaunay triangulation itself allows the definition of a distance measure among regions contained in adjacent planar sections which may be used to control correspondences among regions (the β parameter). From β, our framework allows multiple choices for the connections. Higher values of β cause more regions to be kept in a single connected component, that is, distant regions tend to be connected as β increases. Therefore, the strategy presented here defines a family of possible models derivable from a single set of planar cross sections.

The branching and tiling problems are treated as proposed in the work by Nonato et al. [2001], ensuring that the reconstructed models are piecewise linear manifolds (PL-manifold) and that the intersection of the reconstructed model with the original set of cutting planes results in the original planar regions. These are highly desirable characteristics in many applications such as numerical simulation.

The idea of using numerical parameters to create a family of three-dimensional models is not new and has already been employed in 3D-reconstruction from point clouds, for example, in algorithms such as in the Alpha-shapes by Edelsbrunner and Mücke [1994]. However, the strategy adopted in Alpha-shapes

is not directly applicable to the correspondence problem, and it may also be difficult to ensure that the resulting model is a PL-manifold.

This work is organized as follows. Section 2 presents a brief description of prior work in three-dimensional reconstruction. In Section 3, we review definitions and some properties of the Delaunay triangulation and the Voronoi diagram, necessary to a better understanding of the following section. In Section 4, we describe the β-connection approach, together with some definitions and results on which it is based. The subdivision and disconnection processes employed to solve the branching and tiling problems are presented in Section 5. In Section 6, we present the algorithm and its complexity analysis. Section 7 contains some examples and a discussion of the results. Conclusions and further work are discussed in Section 8.

2. RELATED WORK

In this section, we present a short review of relevant methods for 3D-reconstruction from planar cross sections. In order to contextualize the methodology proposed here, we emphasize the strategy adopted by each method for solving the correspondence problem.

Methods for 3D-reconstruction from planar sections may be classified as implicit, voxel-based, optimal, deformable models and heuristic. Some of them reconstruct the two-dimensional surface that bounds the 3D object, while others generate a volumetric representation comprised of primitive volume elements such as voxels or tetrahedra.

Implicit techniques create an implicit function whose zero set approximates the surface of the reconstructed model. An example of this kind of technique is the work by Jones and Chen [1994]. They define a signed distance field to create an implicit function that, jointly with Marching Cubes [Lorensen and Cline 1987], is used to approximate the boundary surface of the original object. The correspondence is automatically defined by the implicit function, making it difficult to change undesired or inappropriate connections.

Voxel-based models approach a visualization of the reconstructed model directly from the regions contained in the planar sections without building a geometrical surface or volume mesh to represent the object [Kaufman 1998]. Although this kind of strategy can avoid several issues such as presegmentation and those related to the management of the mesh, it is practically impossible to specify and control the branching as the interpolations embedded in the visualization process actually ignore such connections.

Optimal approaches usually use graph theory and optimization strategies to build a 3D model. Early work in this type of reconstruction was developed by Keppel [1975]. In his work, both the correspondence between contours and the surface bounding the maximal volume of the object under reconstruction are obtained from an optimal search in a graph. Later, Shinagawa and Kunii [1991] expanded a discrete toroidal graph to a continuous one. From this continuous graph, it is possible to define parametric surfaces that generate the final model. Homotopy is employed both to construct the parametric surfaces and to define the correspondence among regions. Another optimal reconstruction technique is proposed by Meyers et al. [1992]. They assemble the best-fitting ellipses of each region in order to build a graph from which a minimum spanning tree is derived, determining the correspondences. A final surface is produced by a surface triangulation algorithm.

Deformable models employ geometry, physics, and approximation theory to generate a model from the cross sections. From a seed (initial approximation), deformations are applied seeking to approximate the original object. Although different connections may be obtained using distinct seeds, defining suitable initial elements is not a trivial task. A good overview of this approach can be found in the book by Singh et al. [1998].

A large number of algorithms employ heuristics to construct a 3D model from planar cross sections. One of the first heuristic approaches was that by Christiansen and Sederberg [1978]. It connects contours in neighbouring slices by mapping their bounding box onto a unit square using the shortest edge to make a connection. Ekoule et al. [1991] map the boundaries of the regions on their convex hull and use a shortest edge strategy to connect the convex hulls in adjacent levels. Barequet and Sharir [1996] adopt a geometric hashing method which makes use of orthogonal projections to match and then triangulate portions of contours in adjacent slices. Gaps are filled out by a parametric walking in the regions between two parts of the contours already matched, connecting corresponding points. Bajaj et al. [1996] derive a reconstruction algorithm from a set of properties that should be satisfied by the reconstructed surface model. When the properties cannot be met, the surface is completed by a triangulation that considers the Voronoi skeletons generated from projections. The correspondence among regions is implicitly defined by the surface properties and a volumetric model may also be generated with a tetrahedralization process [Bajaj et al. 1999].

A heuristic approach that inspired many other works in the area is due to Boissonnat [1988]. He builds 2D Voronoi diagrams from the boundaries of regions in consecutive pairs of slices and projects such diagrams to generate a graph from the intersection between the Voronoi edges. From this graph, a 3D Delaunay triangulation is constructed which undergoes a tetrahedron elimination process to generate the volumetric model. Region correspondence is defined both by the 3D Delaunay triangulation and the tetrahedron elimination process. Geiger [1993] improved Boissonnat's approach by handling the graph produced by the projection of the 2D Voronoi diagram in order to handle complex branches and dissimilar contours. Cheng and Dey [1999] use Boissonnat's theoretical results to reconstruct surfaces without generating the 3D Delaunay triangulation. The correspondence problem is also dealt with by projecting Voronoi edges except that projections are restricted to relevant edges only. Contour correspondence is defined by the orthogonal projection of 2D Voronoi edges and by analysing singularities in the model. A difficulty common to all such approaches is the geometric problems involved in computing projections and intersections.

Also in the follow up of Boissonnat's [1988] mathematical framework, Nonato et al. [2001] build a volumetric model from the 3D Delaunay triangulation, adding an approach that guarantees that reconstructed models are 3D manifolds. However, no projections are computed since both tetrahedron elimination and definition of correspondence are handled topologically rather than geometrically. Except for the construction of the initial 3D Delaunay triangulation, the tools developed allow avoiding geometrical computation for the definition of correspondence and branching. Although Nonato's algorithm accepts a previous definition of connections, the correspondence can be automatically determined by a proximity criterion derived from the Delaunay triangulation. It is shown, in that work, that the presence of a special type of tetrahedron, called reverse tetrahedron, indicates that regions in adjacent slices are strongly related and should be connected.

That previous work is extended here to improve the definition of connections amongst regions. The β-connection approach enables a family of models to be constructed from a single set of planar cross sections, allowing multiple choices for the correspondence problem, a feature not found in current reconstruction algorithms. In addition to supporting flexible connections among regions, it preserves the good properties of the previous algorithm, that is, it produces PL-manifolds and meets the resampling condition, guaranteeing that the intersection of the reconstructed model with the original set of cutting planes produces the original planar regions.

3. BASIC CONCEPTS

In this section, we introduce some definitions and properties about the Delaunay triangulation and the Voronoi diagram that will be used in the remaining text.

3.1 Voronoi Diagram and Delaunay Triangulation

Let $A = \{x_1, \ldots, x_n\}$ be a set of points in general position in R^3, that is, there is neither an affine subspace of R^3 containing A nor a subset of A with more than four cospherical points. In the following discussion, the points are always assumed to be in general position.

The *Voronoi diagram* for A is a decomposition of R^3 in convex three-dimensional cells $V_1, \ldots, V_n$ that exhibits the following properties.

(1) Each V_i contains only one point x_i of A;

(2) Given $x \in R^3$, $x \in V_i$ iff $d(x, x_i) \leq d(x, x_j)$ for every $i \neq j$, where d is the euclidean distance.

It can be shown that the intersection of k Voronoi cells, $2 \leq k \leq 4$, is either empty or it is a $(4-k)$-dimensional cell contained in the diagram.

A triangulation, called *Delaunay triangulation*, can be obtained from the Voronoi diagram by associating to each p-dimensional cell of the diagram a $(3-p)$-simplex. This relationship between the Delaunay triangulation and the Voronoi diagram, named duality, is explored by several reconstruction algorithms [Boissonnat 1988; Bajaj et al. 1996; Geiger 1993].

It follows from the duality and the general position hypothesis that each 0-dimensional cell of the diagram is the center of an empty sphere (Delaunay sphere) circumscribing a 3-simplex, that is, the sphere centered in each vertex of the Voronoi diagram does not contain any point of A in its interior. More details on Delaunay Triangulation and Voronoi Diagrams can be found elsewhere [Fortune 1992; Aurenhammer 1991].

When the points of A are distributed in two consecutive parallel planar sections P_1 and P_2 (as in this work), the intersection of the 3D Delaunay triangulation of A with $P_i,\ i = 1, 2$, is the 2D Delaunay triangulation of the points in P_i. This property was proven by Boissonnat [1988] and will be useful in the following sections. Another important fact proven by Boissonnat is that, if the points are vertices of polygons contained in P_1 and P_2, we can ensure that the edges of these polygons are in the Delaunay triangulation even if subdivided.

Based on the above premises, from now on we consider as our input a set of polygons that approximate the boundaries of the regions contained in the parallel planar cross sections. Furthermore, we assume that the edges of these contours are contained in the 3D Delaunay triangulation of its vertices. The polygons satisfying these conditions are called input contours or just contours.

3.2 Voronoi Skeletons and Tetrahedron Types

Let C_1 and C_2 be two sets of contours contained in adjacent planar sections P_1 and P_2 and DT the 3D Delaunay triangulation of the vertices in $C_1 \cup C_2$. Suppose that the contours in C_1 and C_2 are oriented in such a way that the interiors of the regions are always on their left side. Based on the orientation of the contours, an edge of DT contained in P_1 or P_2 can be classified as internal or external based on its position inside or outside the region. With this classification, the contour edges separate the internal and external edges. Note that only the edges contained in $P_i,\ i = 1, 2$, are classified, the edges of DT lying between P_1 and P_2 are not classified.

The tetrahedra of DT either have a face in P_1 (P_2) with an opposite vertex in P_2 (P_1), called *tetrahedra type 1*, or they have one edge in each of the planes $P1$ and $P2$, called *tetrahedra type 2*. A tetrahedron of DT can be classified as internal, external or redundant based on the classification of its edges. A tetrahedron with at least one edge internal to a region r is said to be an *internal tetrahedron* of r. A tetrahedron with at least one external edge and no internal ones is said to be an *external tetrahedron*, and a type 2 tetrahedron with two contour edges is called a *redundant tetrahedron*.

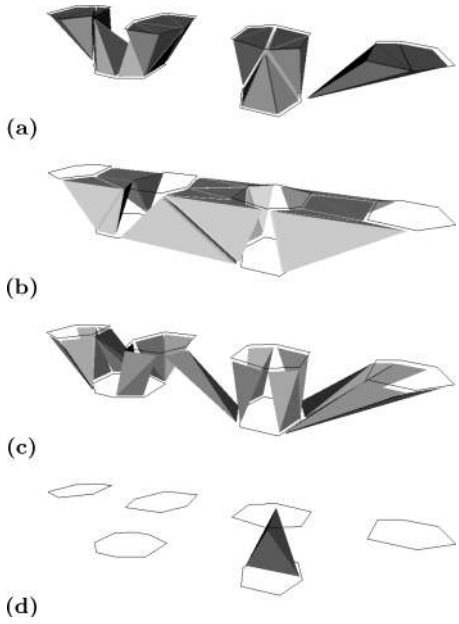

Fig. 2. Types of tetrahedra in the 3D Delaunay triangulation: (a) internal, (b) external, (c) redundant, (d) reverse.

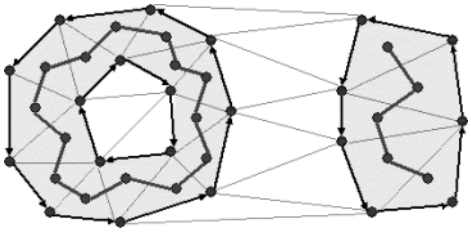

Fig. 3. Voronoi skeleton.

A special case is the *reverse tetrahedron* [Nonato et al. 2001], a type 2 tetrahedron whose edges in P_1 and P_2 are both internal edges. Figure 2 illustrates these definitions showing the different types of tetrahedra. Note that a reverse tetrahedron can be internal to two regions simultaneously. Reverse tetrahedra bear a straight relation with the positioning of regions in adjacent slices as discussed by Nonato et al. [2001].

Let VD_i be the 2D Voronoi diagram of the vertices of C_i in P_i $i = 1, 2$. The *Voronoi skeleton* of a region $r \subset P_i$ is defined as the subset of edges in VD_i that are dual to internal edges of DT whose ends are in the contours of C_i that bound r. Figure 3 shows a set of regions, their contours and correspondent Voronoi skeletons.

We finish this section with a definition that introduces an important concept related to the proximity among regions in adjacent slices, namely, strong overlapping. Strong overlapping gives a measure of proximity between regions and, in Section 4, we show that this measure can be directly obtained from DT.

Definition 3.1. Two regions $r_1 \subset P_1$ and $r_2 \subset P_2$ are strongly overlapping regions if the orthogonal projection of the Voronoi skeleton of r_1 on P_2 intersects the Voronoi skeleton of r_2.

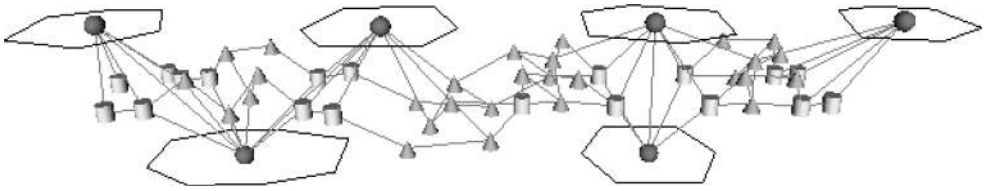

Fig. 4. Graph generated from the Delaunay triangulation. The spheres represent region nodes, whereas cylinders are nodes associated with redundant tetrahedra. The remaining nodes, shown as cones, represent external tetrahedra. External and redundant tetrahedra of the DT are considered external nodes in the graph.

4. BETA-CONNECTION

The β-connetion algorithm tackles the reconstruction problem by handling pairs of slices. Thus, from now on, our discussion is restricted to reconstructing two consecutive parallel planar sections.

The section starts with a description of a graph structure from which β-connection is derived.

4.1 The Graph Associated with DT

Let C_1 and C_2 be two sets of contours representing the boundaries of the regions contained in adjacent planar sections P_1 and P_2, and DT the 3D Delaunay triangulation of the vertices in $C_1 \cup C_2$. From DT, a graph $\mathcal{G}$ can be constructed that has two types of nodes. Each region in P_1 and P_2 and all its internal tetrahedra define a so-called *region node* in $\mathcal{G}$, and each external or redundant tetrahedron of DT define an *external node* in $\mathcal{G}$.

Edges linking pairs of nodes in $\mathcal{G}$ are obtained as follows.

(1) If two regions r_1 and r_2 have internal tetrahedra with a common face, or if there is a reverse tetrahedron with internal edges in r_1 and r_2, then the region nodes representing r_1 and r_2 are connected by an edge in $\mathcal{G}$.

(2) If an external or redundant tetrahedron t shares a face with an internal tetrahedron of a region r, then the region node representing r and the external node representing t are connected by an edge in $\mathcal{G}$.

(3) Two external nodes are connected if and only if the tetrahedra they represent share a face, that is, external and redundant tetrahedra sharing a common face are connected by an edge in $\mathcal{G}$.

Figure 4 illustrates a set of regions and the graph $\mathcal{G}$ generated from DT (see Figure 2). It follows from the above construction process that the graph $\mathcal{G}$ is connected. Futhermore, the degree of any external node in $\mathcal{G}$ ranges from 1 to 4.

A path of length n connecting two region nodes a and b in $\mathcal{G}$ is a sequence of nodes $\{\tau_1, \ldots, \tau_{n+1}\}$, such that $\tau_1 = a$, $\tau_{n+1} = b$, and (τ_i, τ_{i+1}), $i = 1, \ldots, n$, is an edge of $\mathcal{G}$.

Definition 4.1. The distance between two region nodes a and b in $\mathcal{G}$, denoted $d_{\mathcal{G}}(a, b)$, is the length of the shortest path connecting a and b.

As we show in the following, graph $\mathcal{G}$ provides understanding of the behavior of DT regarding the connection of regions. Moreover, it offers a sound framework for deriving the β-connection algorithm.

4.2 The Beta-Components

This section introduces the concept of β-components which is the basis of the β-connection algorithm.

Definition 4.2. Let β be a natural number, that is, β can assume values from the set $\{0, 1, 2, \ldots\}$. Two region nodes a and b are said to be β-connected, denoted $a \overset{\beta}{\approx} b$, if there is a sequence of region nodes $\{\sigma_1, \ldots, \sigma_k\}$ in $\mathcal{G}$, where $\sigma_1 = a$ and $\sigma_k = b$, such that $d_{\mathcal{G}}(\sigma_i, \sigma_{i+1}) \leq \beta$.

LEMMA 4.3. $\overset{\beta}{\approx}$ *is an equivalence relation.*

PROOF. Let a, b, and c be three region nodes in $\mathcal{G}$ such that $a \overset{\beta}{\approx} b$ and $b \overset{\beta}{\approx} c$. Since $d_{\mathcal{G}}(a,a) = 0$, $a \overset{\beta}{\approx} a$, and the relation is reflexive. The symmetry is evident from the definition of $d_{\mathcal{G}}$ once $d_{\mathcal{G}}(a,b) = d_{\mathcal{G}}(b,a)$. Finally, as long as $a \overset{\beta}{\approx} b$ and $b \overset{\beta}{\approx} c$, there are two sequences of region nodes $\{\sigma_1, \ldots, \sigma_k\}$ and $\{\tau_1, \ldots, \tau_m\}$ where $\sigma_1 = a, \sigma_k = b, \tau_1 = b, \tau_m = c$ and $d_{\mathcal{G}}(\sigma_i, \sigma_{i+1}) \leq \beta$, $d_{\mathcal{G}}(\tau_j, \tau_{j+1}) \leq \beta$. Therefore, the distance $d_{\mathcal{G}}$ between consecutive pairs in the sequence $\{a, \sigma_2, \ldots, \sigma_{k-1}, b, \tau_2, \ldots, \tau_{m-1}, c\}$ is smaller than, or equal to β, proving the transitivity of $\overset{\beta}{\approx}$. □

As each region node represents a region contained in a plane P_i, $i = 1, 2$, β also defines equivalence classes for the original regions, thus allowing the correspondence to be specified through equivalence classes. Therefore, different values of β can produce different equivalence classes, enabling multiple choices of correspondence among the regions.

Definition 4.4. The β-components of a model to be reconstructed are defined by the contours within the same equivalence class generated from β.

An interesting characteristic of the this process is that a value $\beta = 0$ implies no correspondence among regions, and, as β increases, the number of regions within a particular equivalence class also tends to increase until all regions are contained in a single class. The next proposition presents an important property of β-connection, namely, the proximity relationship between strongly overlapping regions.

PROPOSITION 4.5. *Let $r_1 \cup P_1$ and $r_2 \cup P_2$ be two strongly overlapping regions, then r_1 and r_2 are in the same equivalence class for any $\beta \geq 1$.*

PROOF. As r_1 and r_2 are strongly overlapping regions, the orthogonal projections of their Voronoi skeletons intersect. Let v_1 and v_2 be the skeleton edges of r_1 and r_2 intercepted by the skeleton projection. If d_1 and d_2 are internal Delaunay edges dual to v_1 and v_2, it is not difficult to see that the intersection of the sphere defined by the ends of d_1 and d_2 with P_1 and P_2 produces empty circles in these planes. Thus, d_1 and d_2 define a reverse tetrahedron in DT , ensuring that $\mathcal{G}$ has an edge linking the region nodes that represent r_1 and r_2. Therefore, for any $\beta \geq 1$, r_1 and r_2 will be in the same equivalence class. □

From Proposition 4.5 we can see that strongly overlapping regions share reverse tetrahedra, guarateeing that they are always the first ones to be grouped when parameter β increases from zero. It can also be shown ([Nonato et al. 2001]) that reverse tetrahedra appear only between strongly overlapping regions. Thus, DT offers a topological mechanism to identify region proximity, given by the presence (or absence) of reverse tetrahedra connecting the regions.

Another interesting property of DT is that it provides, through $\mathcal{G}$, a measure of distance among regions in consecutive planar sections. When two contours in adjacent slices are shifted apart as shown in Figure 5, Delaunay empty spheres touching both contours tend to appear in the space between them, originating a barrier of redundant tetrahedra and increasing the distance $d_{\mathcal{G}}$ and the value of β needed for keeping them in the same equivalence class. Figures 5(b) and 5(c) illustrate this property of DT.

The previous discussion implies that the 3D Delaunay triangulation itself introduces a type of distance measure among regions contained in adjacent planar cross sections. This distance can be used as an input parameter to define a family of models from a given set of slices.

Other authors [Boissonnat 1988; Cheng and Dey 1999; Geiger 1993] have already suggested that the 3D Delaunay triangulation bears a proximity measure among regions in adjacent slices. To the best of our knowledge this is the first time, however, that such a measure is explicitly shown.

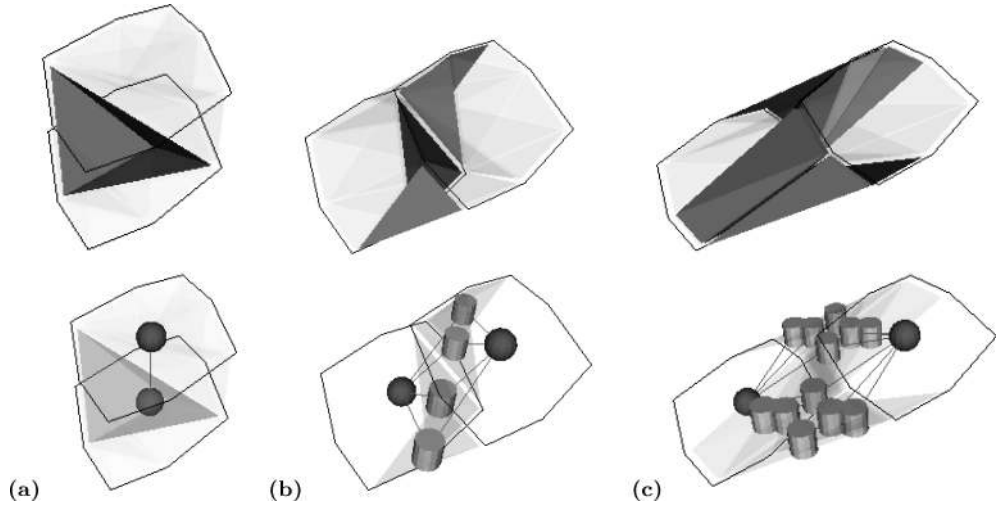

Fig. 5. The Delaunay triangulation creates a barrier of redundant tetrahedra when the contours are shifted apart. These figures show the redundant tetrahedra and the graph $\mathcal{G}$.

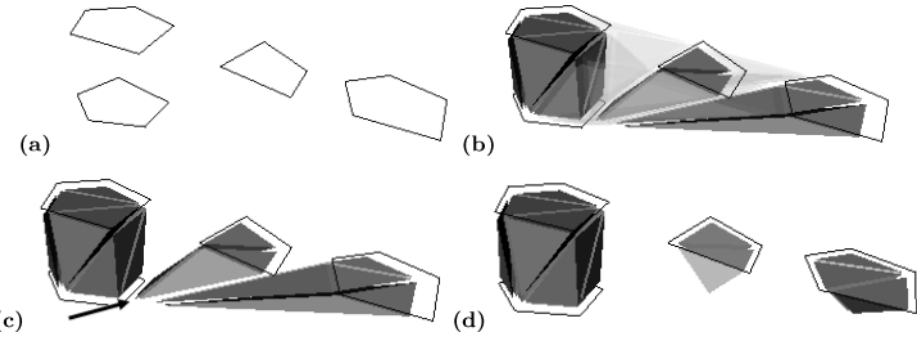

Fig. 6. (a) Set of contours; (b) Delaunay triangulation (external tetrahedra are transparent) (c) external tetrahedra elimination; (d) vertices displacement.

5. COMPONENT DISCONNECTION AND SUBDIVISION PROCESS

A natural way to define the connected components of the reconstructed model is to associate them with the β-components. Since the outcome of the Delaunay triangulation is a single, connected, simplicial complex, the tetrahedra in DT linking components that should not be connected (because they are not in the same β-components) must be removed. This tetrahedron elimination process is described in the following.

5.1 Generation of Connected Components

Suppose that the connected components have already been specified through the equivalence classes defined by the β parameter. In order to disconnect those components that belong to different β-components, it is necessary to remove from DT all external and redundant tetrahedra whose vertices are not in the same component.

Although tetrahedron elimination is necessary, it may not be sufficient to disconnect the components. In some situations, as illustrated in Figure 6, components can remain connected through vertices or

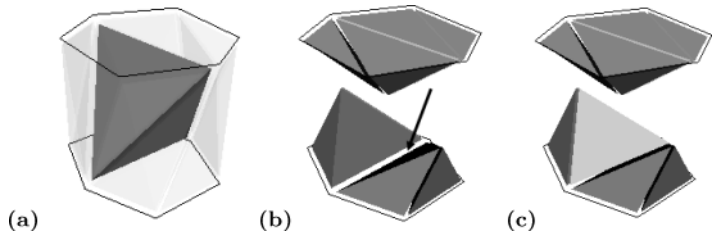

Fig. 7. (a) Reverse tetrahedron; (b) singularity produced by displacing vertices; (c) reconstruction after duplicating the reverse tetrahedron.

contour edges even after tetrahedra elimination. In this case, vertices (and edges) must be translated to disconnect components completely, as shown in Figure 6(d). Such a translation is implemented in the following manner. First, the centroids of each connected component that will have elements moved are computed. The centroid of a component is calculated from the vertices of its associated contours. The offending elements are then displaced towards the centroid, keeping their z coordinate (assuming z axis orthogonal to the slices) at a predefined intermediate position level between two slices. In our implementation that level was set at one fourth of the interslice distance.

$\beta = 0$, that causes each region to generate an independent connected component, is a case that deserves special attention. In this case, the reverse tetrahedra connecting strongly overlapping regions do not allow disconnection to be executed as stated earlier. The problem is that the vertex displacement employed to disconnect regions maintains the reverse tetrahedra attached to only one region, generating wedges in the model. Figures 7(a) and 7(b) illustrate this scenario.

A wedge can originate a singularity, that is, its neighborhood is homeomorphic neither to a sphere nor to a half-sphere. Singularities must be avoided to ensure that the reconstructed model is a PL-manifold. Thus, to disconnect strongly overlapping regions, it is necessary to duplicate the reverse tetrahedra and then translate duplicate vertices and edges to an appropriate intermediate position, as shown in Figure 7(c).

5.2 Subdivision of Tetrahedra

After the disconnection process, the external tetrahedra in each component must also be eliminated to guarantee the resampling condition. However, as in the component disconnection case, this may introduce singularities. To ensure the resampling and manifold conditions in the reconstructed model, a tetrahedron subdivision process is employed. The subdivision strategy adopted here is similar to the one described in detail by Nonato et al. [2001], summarized as folows.

(1) Each external tetrahedron whose elimination does not generate singularities is removed.

(2) The remaining external tetrahedra are subdivided.

(3) The new vertices introduced by the subdivision are displaced to an intermediate position between the consecutive slices.

The subdivision must respect two criteria. For a type 1 tetrahedron, a new vertex is inserted in each external edge. The number of new tetrahedra depends on the number of new vertices created: two new tetrahedra are created if one vertex is added, and three or four new tetrahedra are created if two or three vertices are added, respectively. Figure 8(a) shows the subdivision cases for a type 1 tetrahedron. For a type 2 tetrahedron, two or four new tetrahedra may be introduced, depending on whether one or two vertices were inserted into the external edges, as shown in Figure 8(b).

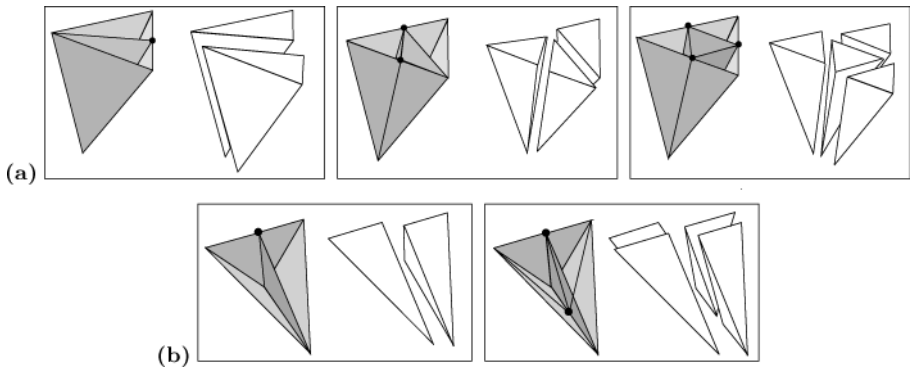

Fig. 8. Subdivision cases for: (a) type 1 tetrahedron; (b) type 2 tetrahedron.

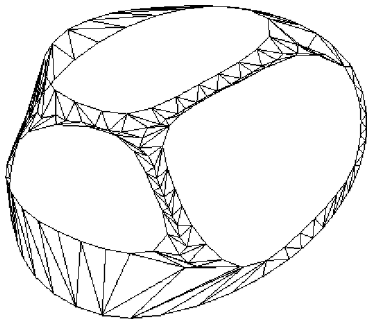

Fig. 9. Result of reconstruction after tetrahedron subdivision in a one-to-three branching case.

In addition to guaranteeing the resampling and manifold conditions, the subdivision handles the branching problem properly, as illustrated in Figure 9, where one contour must be connected to three other ones. For further information, we refer to a previous work by Nonato et al. [2001] where handling of subdivision and branching is treated and illustrated in detail.

In the following section, we describe how the previously described procedures are integrated in the β-Connection algorithm.

6. β-CONNECTION ALGORITHM

The β-connection algorithm can be outlined as follows.

Let C be a set of contours bounding the regions contained in two adjacent planar sections and β be a natural number.

(1) *Delaunay triangulation*. Compute the 3D Delaunay triangulation of vertices in C. Classify tetrahedra as internal, external, reverse, or redundant as outlined in Section 3.2.
(2) *β-components*. Create the graph associated to DT and compute the β-components from the equivalence classes defined by β, as outlined in Section 4.
(3) *Connected components*. Remove the external and redundant tetrahedra connecting the β-components. If necessary, displace vertices and duplicate reverse tetrahedra, as outlined in Section 5.
(4) *Reconstruction*. For each β-component (see Section 5):
   (a) remove the external tetrahedra whose elimination does not introduce singularities (and the redundant tetrahedra attached to them);
   (b) subdivide the remaining external tetrahedra;
   (c) translate the new vertices to an intermediate position between slices.

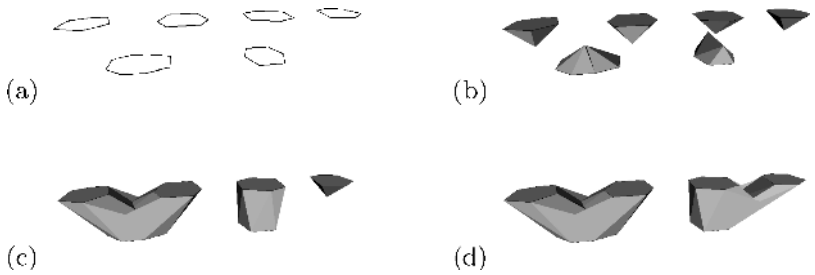

Fig. 10. (a) Original contours; models generated for β equal: (b) 0; (c) 1; (d) 3.

A few issues must be highlighted regarding the implementation of the algorithm.

—The distances and equivalence classes that completely define the connected components may be extracted directly from the Delaunay triangulation produced in Step 1. Thus, it is not necessary to construct an explicit representation of its associated graph.

—In order to implement tetrahedral elimination in Step 3, all external and redundant tetrahedra with vertices in different components are deleted. After removal, vertices from one component that remained in the other one are the ones to be displaced as illustrated in Figure 6.

—In Step 4, to verify whether tetrahedral elimination introduces singularities, the stars of the vertices lying on the external tetrahedra of a component are checked. If verification indicates that no singularities will be introduced, the tetrahedra are eliminated. Otherwise, they are subdivided.

The algorithm performs geometrical calculations only when DT is being built and later in the vertices displacement process. All the remaning operations and tests are topological, thus ensuring a robust and efficient implementation.

Since the classification, elimination, and subdivision steps can all be done in linear time on the number of tetrahedra, the computational cost of the algorithm is governed by the construction of the Delaunay triangulation and by the computation of the β-components. The 3D Delaunay triangulation can be built in $O(n^2)$ with an incremental algorithm [Fortune 1992] where n is the number of vertices in the triangulation. Computation of β-components is reduced to computing, for each region node $\sigma_i \in \mathcal{G}$, the distance from σ_i to all region nodes $\sigma_j \in \mathcal{G},\ i \neq j$. This computation can be done with a breadth-first search algorithm [Cormen et al. 1997] on graph $\mathcal{G}$ in $O(N_r E)$, where N_r and E are the numbers of region nodes and edges in $\mathcal{G}$, respectively.

7. RESULTS AND EXAMPLES

This section presents some models created with the β-connection algorithm. The code was written in C++ and uses the CGAL library (`www.cgal.org`) to build the 3D Delaunay triangulation. Visualizations were generated with the Visualization Toolkit (VTK) [Schröeder et al. 2002].

We start with a simple but illustrative example that shows how the β-parameter defines the correspondences. Figure 10(a) presents a set of six contours, two in one slice and four in the following one, from which 3D-models are to be reconstructed. Figures 10(b) to 10(d) show the models produced by β-connection with values of β equal to 0, 1 and 3, respectively. As previously mentioned, higher values of β cause more regions to be connected. Varying β thus enables multiple reconstruction alternatives to be considered.

The flexibility of establishing alternative connections renders β-connection an interesting and powerful tool in solid modeling applications, because varying a single discrete parameter allows a whole family of models to be constructed from the same set of contours. Figures 11 and 12 illustrate this

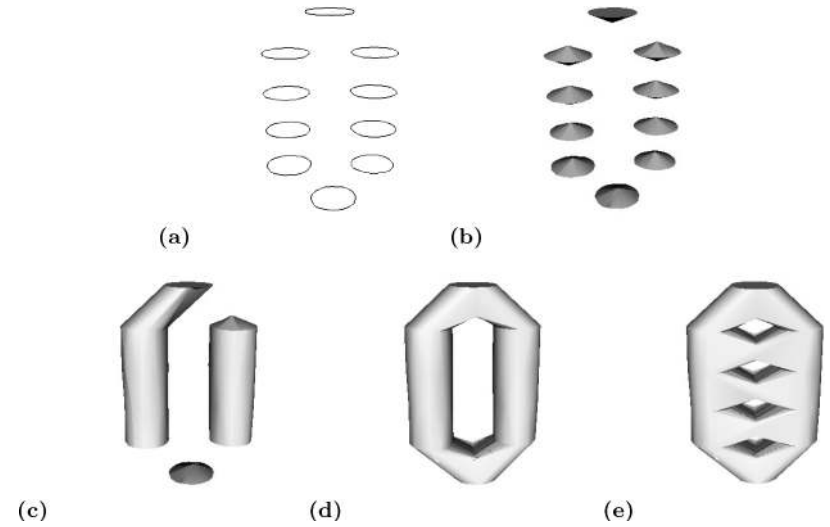

Fig. 11. Models generated from the countours in (a) for β equal to: (b) 0; (c) 1; (d) 3; (e) 5.

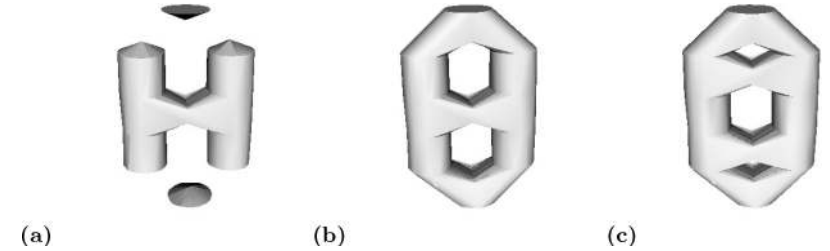

Fig. 12. Models generated with different values of β in each level. The sets of values are (a) $(0, 1, 5, 1, 0)$; (b) $(2, 1, 5, 1, 2)$; (c) $(2, 5, 1, 5, 2)$.

potential. Figures 11(b) to 11(e) present three different models generated from the contours shown in Figure 11(a) for β values equal to 0,1,3 and 5, respectively.

Another truly useful property of β-connection that makes it a versatile tool in solid modeling applications is the possibility of varying the values of β across different pairs of slices. This is possible because the algorithm separately handles regions contained in pairs of adjacent planar sections. Figures 12(a) to 12(c) show three different models obtained by assigning different β values to each pair of slices. One can observe how the topology of the resulting models changes in each case.

We illustrate the potential applicability of the approach for modeling objects in pratical situations with an example from real data. Figure 13(a) shows a set of contours lying in planar cross sections of a blood vessel. In complex structures such as this one, it is very difficult to decide the correct way to connect the regions. Figures 13(b) and 13(c) present the models produced from the contours in Figure 13(a) for β equal to 1 and 2, respectively. One may compare Figure 13(a) and 13(b), observing how the connections changed. For example, those elements pointed by the black arrows remain disconnected for a choice of $\beta = 1$, but are connected together when β assumes the value 2.

In addition to the elements visually indicated in Figure 13(b), several other regions are connected differently when β increases from 1 to 2. Figures 14(a) and 14(b) show a wireframe view of part of the blood vessel model to illustrate how the connections changed.

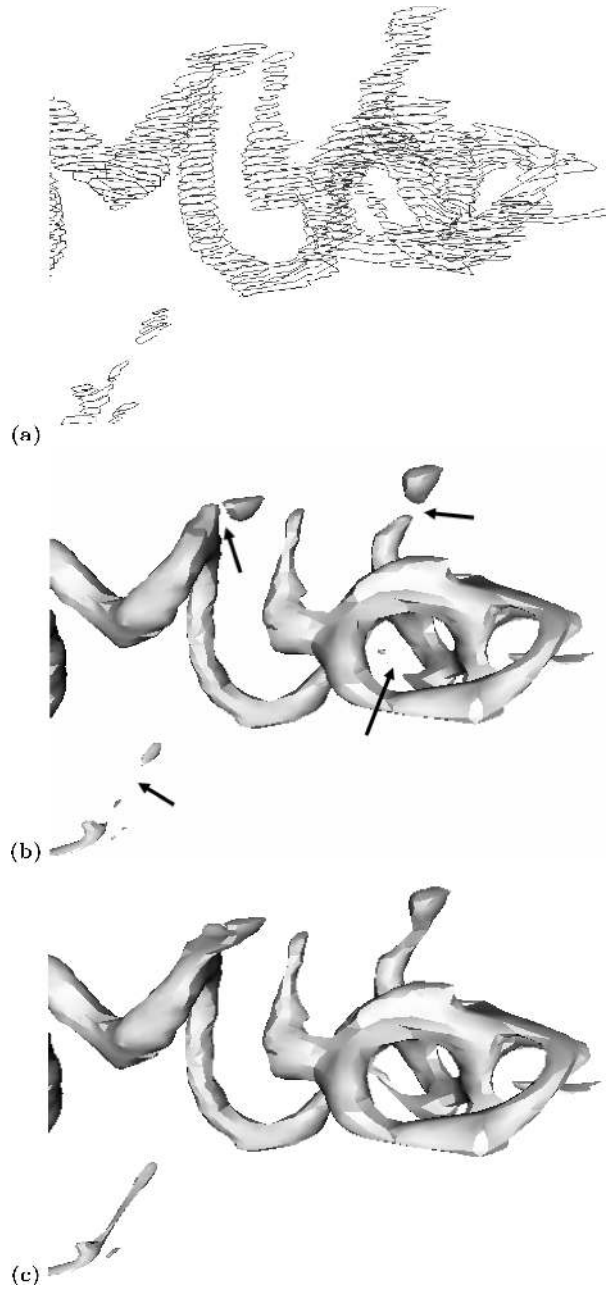

Fig. 13. (a) A set of contours of a blood vessel; (b) model generated for $\beta = 1$; (c) model generated for $\beta = 2$.

Figure 15 shows the complete reconstruction of the whole set of 151 slices of the blood vessel. It illustrates both the algorithm capabilities for handling large data sets with complex geometry, and how the β parameter may be tuned to derive a model that better approximates the real object.

Highlighted in Figure 15(b) are the regions where the change of β made the most difference. The region highlighted in the top right rectangle corresponds to the same blood vessel segment displayed

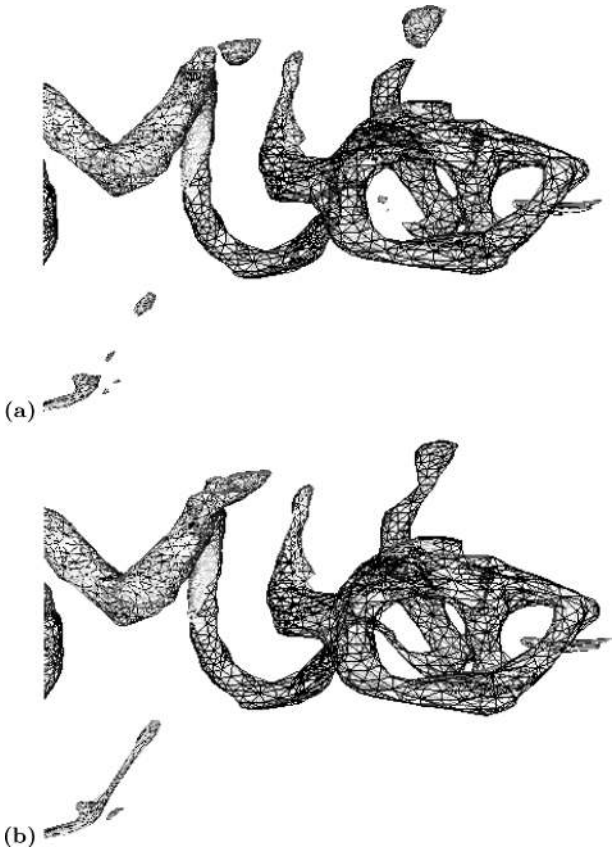

Fig. 14. Wireframe viewing of the reconstructed blood vessel. (a) $\beta = 1$; (b) $\beta = 2$.

in Figures 13 to 14. Figure 15(d) superimposes one reconstruction on the other to emphasize the parts with most changes.

Both reconstructions for the other two highlighted regions are shown in detail in the following figures. Figure 16 shows zoomed views of the region in the bottom right rectangle in Figure 15. Figure 17 shows the reconstructions of the region in the top left rectangle in Figure 15 for two values of β.

Table I shows the execution times for each step of the reconstruction process applied to the blood vessel data whose resulting model is shown in Figure 15. The data contains 151 slices with 1,222 contours and 29,141 vertices. The boundary of the resulting mesh contains 55,260 triangles for $\beta = 1$ and 55,286 triangles for $\beta = 2$. Execution times show that the largest computational cost is due to the Delaunay triangulation. Tetrahedral elimination and handling of external tetrahedra within components, called branching and concavity in the table, are the next most expensive steps. The table ignores the time spent to load and write data onto a disk.

It is worth noting that, in applications requiring simulations with different models, such as surgical planning and fluid flow, the ability of redefining the correspondences and investigating alternatives is paramount. Existing reconstruction techniques do not offer an effective mechanism to investigate alternative correspondences, and the task can thus become tiresome and time-consuming. β-connection provides a straightforward approach for changing the correspondences and generating alternative models for further investigation.

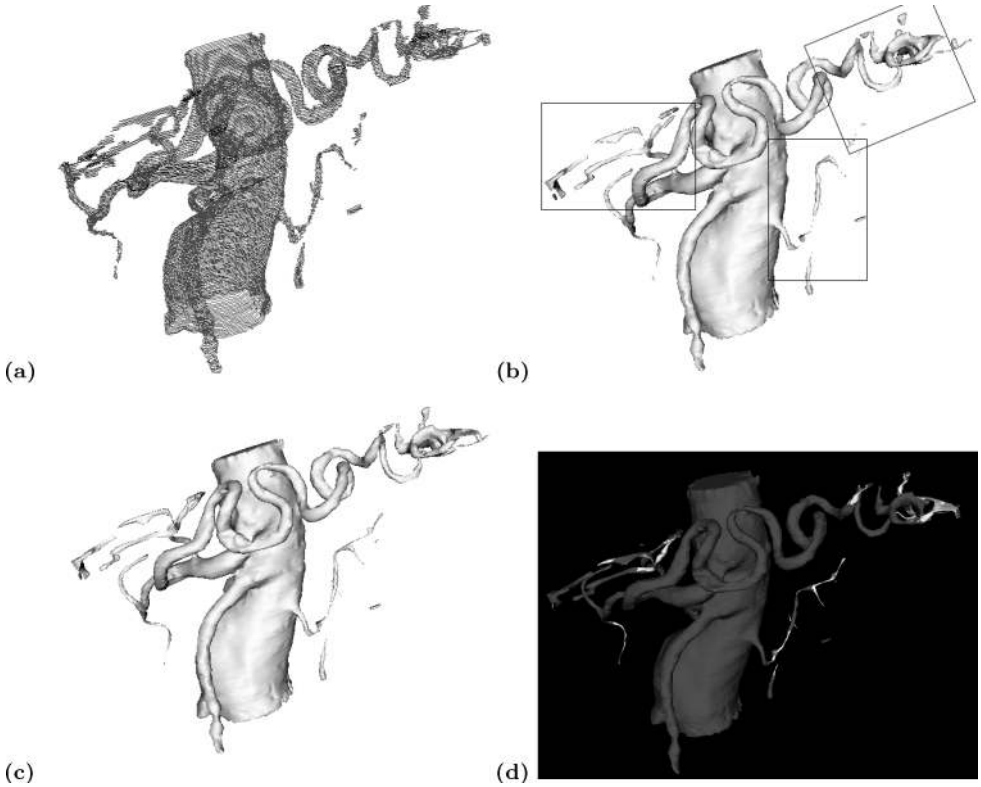

Fig. 15. Views of the reconstruction of the whole blood vessel. (a) contours; (b) $\beta = 1$; (c) $\beta = 2$; (d) superimposes one reconstruction on the other.

8. CONCLUSIONS AND FUTURE WORK

This article introduces a novel approach to treat the connection problem in the reconstruction of 3D-structures from planar cross sections. The technique offers great flexibility in the choice of how components are to be connected and solves the branching and tiling problems in a very satisfactory way.

From a theoretical framework that allows inferring a distance measure from the 3D Delaunay triangulation constructed from a set of planar regions, our methodology calculates the distance among regions, and an algorithm is defined that can generate a one-parameter family of models from a given set of samples.

The algorithm's applicability was illustrated with the generation of solid models with different topologies from a single set of contours and the reconstruction of an arterial system from real data. The possibility of varying the value of parameter β to produce different results in the reconstruction allows a user to investigate alternative types of connections that would not be easily obtained otherwise and, thus, to study the nature of certain objects closely. Moreover, the fact that the resulting mesh is free of singularities enables the resulting models to be used in numerical simulations of real physical phenomena with no need for postprocessing.

A major contribution of this technique is to ease the user's intervention on the decision of the connections as it can now be made by varying a single parameter (β), rather than by acting directly on the contours which is a main drawback of reconstruction from planar cross sections. The usability of the reconstruction process is, therefore, largely increased.

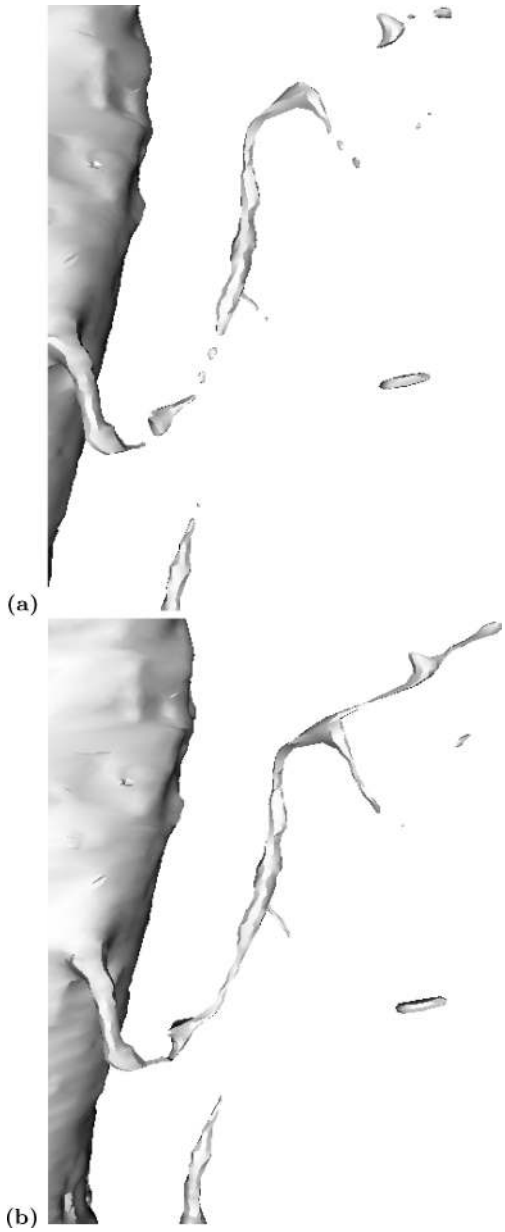

Fig. 16. Detail on bottom right corner of Figure 15. (a) $\beta = 1$; (b) $\beta = 2$.

This facility may be particularly useful when studying certain data sets for which connections are an important issue but are not completely understood. One example are studies of water infiltration in soil for agricultural purposes. Researchers now can track the path followed by water in a soil sample from a set of 2D-tomographic images. Segmentation, 3D-reconstruction, and visualization of water paths in the soil can assist domain specialists in conceiving and studying models of soil behavior in this context. Applications of this type also motivate our current work.

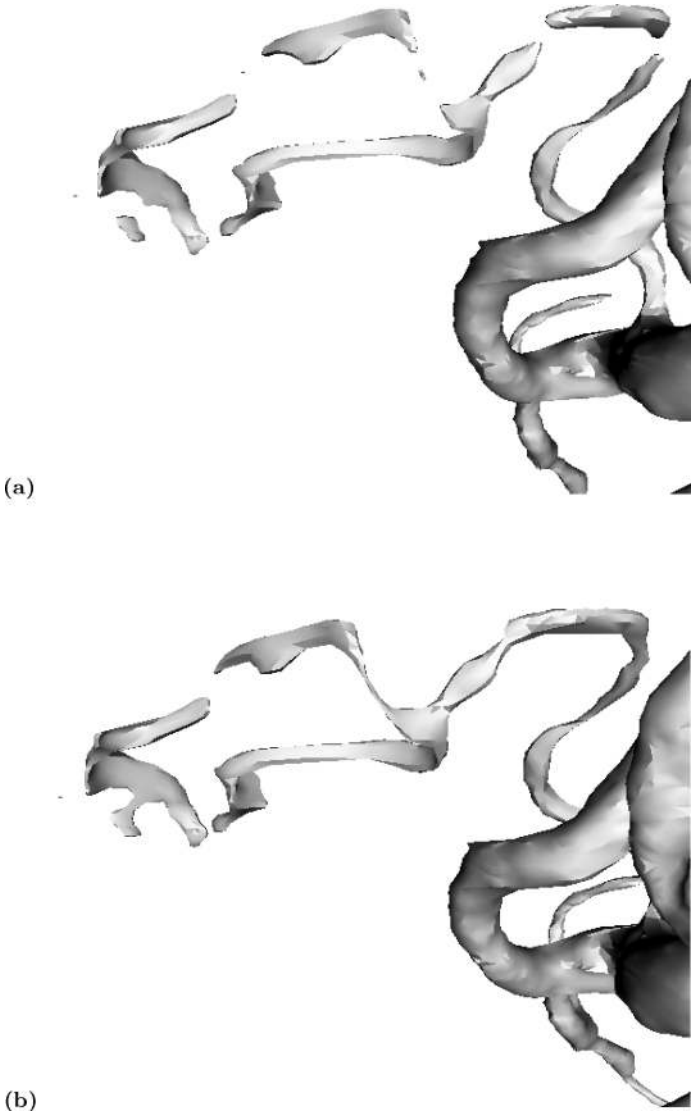

Fig. 17. Detail on top left corner of Figure 15. (a) $\beta = 1$; (b) $\beta = 2$.

We are now working to improve the quality of the volumetric mesh generated by β-connection so that it becomes better suited for numerical simulation. An overall remeshing technique, applicable to the meshes produced by the algorithm, is under development.

Further work in our plans is to compare the variation of parameter β against the correctness of the resulting models. Taking as a starting point a set of known objects with varying topology, we intend to relate the sampling resolution with the β value required to obtain the correct object as output, hoping to achieve a method that will automatically decide the ideal β value for a particular class of objects. For instance, taking a typical blood vessel, or a typical soil sample of some kind, we wish to define what β values could be used to ensure a correct reconstruction at a given sampling rate.

Table I. Computational Times (in Ms and Percentual Values) for Each Step of the Reconstruction Process

	3D Delaunay Triangulation	Edge Classification	Tetrahedra Classification	Connected Components
$\beta = 1$	228.20 74.11%	0.50 0.16%	0.51 0.16%	8.78 2.85%
$\beta = 2$	228.19 74.11%	0.53 0.17%	0.47 0.15%	8.45 2.74%

	Tetrahedra Elimination	Displacement of Vertices (Edges)	Branching and Concavities	Tetrahedra Subdivision
$\beta = 1$	33.40 10.85%	2.01 0.65%	34.37 11.16%	0.06 0.03%
$\beta = 2$	33.23 10.79%	2.08 0.67%	35.82 11.64%	0.14 0.05%

ACKNOWLEDGMENT

We are grateful to Dr. G. Barequet from the Technion Israel Institute of Technology for making available the blood vessel data used to generate the figures in the results section.

REFERENCES

AURENHAMMER, F. 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv. 23*, 3, 345–405.

BAJAJ, C., COYLE, E., AND LIN, K. 1996. Arbitrary topology shape reconstruction from planar cross sections. *Graph. Models Image Process. 58*, 6, 524–543.

BAJAJ, C., COYLE, E., AND LIN, K. 1999. Tetrahedral meshes from planar cross-sections. *Comput. Meth. Appl. Mech. Eng. 179*, 31–52.

BAREQUET, G. AND SHARIR, M. 1996. Piecewise-linear interpolation between polygonal slices. *Comput. Vision Image Understand. 63*, 2, 251–272.

BOISSONNAT, J.-D. 1988. Shape reconstruction from planar cross sections. *Comput. Vision Graph. Image Process. 44*, 1–29.

CHENG, S.-W. AND DEY, T. 1999. Improved constructions of delaunay-based contour surfaces. In *ACM Symposium on Solid Modeling and Applications*. ACM Press, New York, NY. 322–323.

CHRISTIANSEN, H. AND SEDERBERG, T. 1978. Conversion of complex contour line definitions into polygonal element mosaics. *Comput. Graph. 12*, 3, 187–192.

CORMEN, T., LEISERSON, C., AND RIVEST, R. 1997. *Introduction to Algorithms*. Electrical Engineering and Computer Science Series. MIT Press, Cambridge, MA.

EDELSBRUNNER, H. AND MÜCKE, E. P. 1994. Three-dimensional alpha shapes. *ACM Trans. Graph. 13*, 1, 43–72.

EKOULE, A., PEYRIN, F., AND ODET, C. 1991. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. Graph. 10*, 2, 182–199.

FORTUNE, S. 1992. Voronoi diagrams and delaunay triangulation. In *Computing in Euclidean Geometry*, D.-Z. Du and F. Hwang, Eds. Lecture Notes Series on Computing, Vol. 1. World Scientific Publishing Company, Singapore, 193–233.

GEIGER, B. 1993. Three dimensional modeling of human organs and its application to diagnosis and surgical planning. Tech. rep. 2105, INRIA, Sophia-Antipolis, France.

JONES, M. AND CHEN, M. 1994. A new approach to the construction of surfaces from contour data. *Comput. Graph. For. 13*, 3, 75–84.

KAUFMAN, A. 1998. Volume visualization: Principles and advances. In *SIGGRAPH'98*, Course Notes 24.

KEPPEL, E. 1975. Approximating complex surface by triangulation of contour lines. *IBM J. Res. Devel. 19*, 2–11.

LORENSEN, W. AND CLINE, H. 1987. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Comput. Graph. 21*, 163–169.

MEYERS, D., SKINNER, S., AND SLOAN, K. 1992. Surfaces from contours. *ACM Trans. Graph. 11*, 3, 228–258.

NONATO, L., MINGHIM, R., OLIVEIRA, M., AND TAVARES, G. 2001. A novel approach for Delaunay 3d reconstruction with a comparative analysis in the light of applications. *Comput. Graph. For. 20*, 2, 161–174.

SCHRÖEDER, W., MARTIN, K., AND LORENSEN, W. 2002. *Visualization Toolkit, An Object-Oriented Approach to 3D Graphics 3rd Ed*. Kitware Inc., New York, NY.

SHINAGAWA, Y. AND KUNII, T. 1991. The homotopy model: A generalized model for smooth surface generation from cross sectional data. *Visual Comput. 7*, 72–86.

SINGH, A., GOLDGOF, D., AND TERZOPOULOS, D., EDS. 1998. *Deformable Models in Medical Image Analysis*. IEEE Computer Society Press.

Received August 2003; revised April 2004; accepted July 2005