1995

Arbitrary Topology Shape Reconstruction from Planar Cross Sections

Chandrajit L. Bajaj

Edward J. Coyle

Kwun-Nan Lin

Report Number:

95-027

ARBITRARY TOPOLOGY SHAPE RECONSTRUCTION FROM PLANAR CROSS SECTIONS

Chandrajit L. Bajaj
Edward J. Coyle
Kwun-Nan Lin

CSD-TR-95-027
April 1995

Arbitrary Topology Shape Reconstruction from Planar Cross Sections

Chandrajit L. Bajaj Edward J. Coyle Kwun-Nan Lin

Department of Computer Science,
Purdue University,
West Lafayette, IN 47907

School of Electrical Engineering,
Purdue University,
West Lafayette, IN 47907

email: {bajaj@cs, coyle@ecn klin@cs}.purdue.edu

Abstract

Existing algorithms in reconstructing surfaces from image volumes have disadvantages. The volume-based approaches, such as the marching cubes, are robust on different topologies but generate a tremendous number of unordered triangles. The surface-based approaches generate a much lower number of triangles, but they have limitations with complicated topologies. We present a new surface based approach. We define three criteria for the desired reconstructed surfaces to correspond well to the actual physical model and consequently to result in a natural looking surface rendering. Precise corresponding and tiling rules are derived from these criteria. These rules enforce proper tiling between contours on adjacent slices. Holes, branching regions, and dissimilar areas of contours cannot be tiled without breaking these rules. Finally, the un-tiled regions are tiled with their medial axis. Given any input data, our algorithm guarantees that the reconstructed surfaces satisfy the desired surface criteria. We develop a new tiling approach to achieve near optimal tiling even in the branching regions. We fully implement our algorithm. It inputs either a set of contour data or a volume of image slices and generates triangles of 3D iso-surfaces without any user intervention. The results of actual medical data are presented. We conclude that the features of robustness, good correspondence to actual physical models, compact surface representation, and automation make our algorithm valuable in surface reconstruction.

1 Introduction

Meyers et al. [19] categorize the surface reconstructing algorithms to volume based and surface based. The volume based methods divide the image volume into small voxels and calculate the iso-surface in each voxel. A typical example is the marching cubes approach [18]. The volume based approaches usually generate a tremendous number of triangles. For example, it is not unusual for a volume of 100 256*256 CT slices to generate more than half million triangles. The geometrically deformed models (GDB) of Miller et al. [20] uses relaxation to grow a solid model to fill the iso-surface. It generates a lower number of triangles compared to the marching cubes approach. The surface based approaches first identify iso-contours from each slice and then reconstruct iso-surfaces from these contours. They generate much fewer triangles compared to the volume based approaches because the flatter regions are represented by larger triangles. But, as described by Meyers et al.[19], the surface based approaches suffer from the correspondence problem, the tiling problem, and the contour branching problem.

1.1 Correspondence problem

The correspondence problem is to find the correct connections between the contours of adjacent slices. Fig. 1 shows an example with four different join topologies (b)-(e) resulting from the same cross sections as in (a). If the distance between slices is large, a priori-knowledge or global information is required to determine the correct correspondence. Bresler et al.[3] use domain knowledge to reduce the problem. Meyers et al.[19] and Soroka [25] approximate the contours by eclipses and then assemble them into cylinders to determine the correspondence. Wang et al. [26] check the overlapping area.

1.2 Tiling and branching problems

Tiling means to triangulate the strip formed between contours of the adjacent slice (Fig. 6(a)). There are two related issues. One is how to accomplish optimal tiling in terms of certain metrics such as triangle areas, enclosed volumes, etc. The other is the topological correctness of tiling. Gitlin et al. [12]

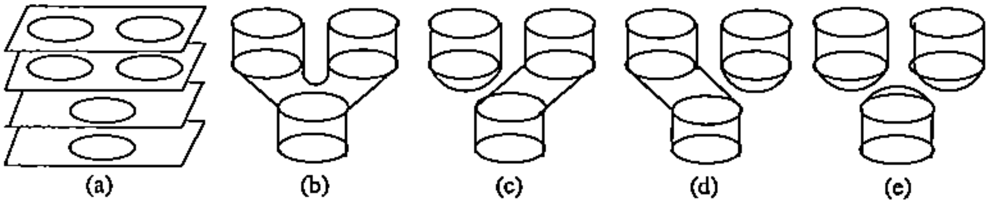


Figure 1: Correspondence problem: (a) the cross section contours, (b)-(e) show that different topologies have the same cross sections as in (a).

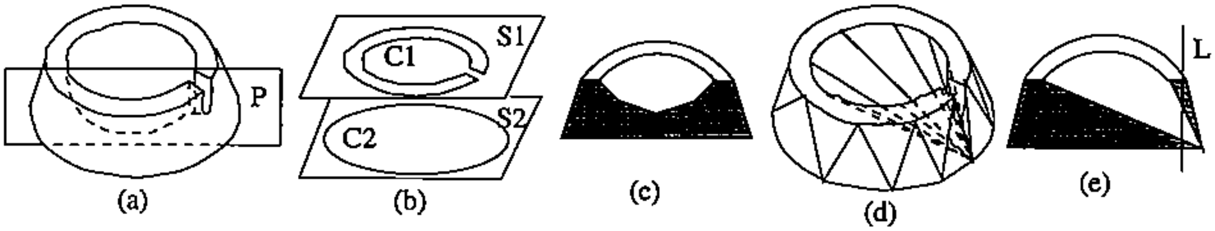


Figure 2: Dissimilar contours: (a) the model, (b) the contours on two slices, (c) the vertical section along the plane *P* in (a), (d) incorrect tiling, (e) wrong vertical section resulted from (d).

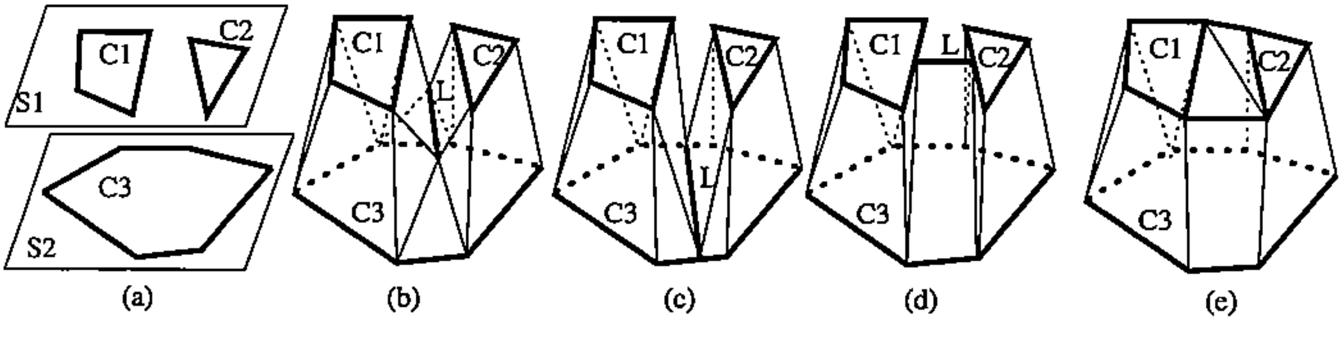


Figure 3: Different types of branching contours processing: (a) branching contours, (b)-(e) different handling of branching.

show one example that two polygons cannot be connected to form a simple polyhedron. When two corresponding contours are very different, it is difficult to obtain a topologically correct tiling. For example, Fig. 2(a) and (b) show a solid model and its cross-section contours. The model is formed by digging a shallow hole and a ditch on the top surface. Fig. 2(c) shows the vertical cross section along the plane P in (a). If the interior of the top contour is tiled to the bottom contour as in (d), the cross section of the reconstructed model along plane P is shown in (e). It is unnatural because the scalar data along line L flips its sign twice between slices. The interior region of the contour $C1$ of (b) should not be tiled to the bottom contour $C2$. This example illustrates the difficulty of obtaining topologically correct tiling even when there is no branching.

The branching problem occurs when one contour corresponds to more than one contour in an adjacent slice. Fig. 3(a) shows that contour $C3$ on the bottom slice branches into $C1$ and $C2$ on the top slice. A contour having no correspondence forms a hole. The branching complicates the tiling, and it also creates the problem of branching surface reconstruction.

The problem of matching points between contours was formalized by Keppel [15] into a graphic search problem. Fuchs et al [9] provide an efficient algorithm based on an Euler tour of a toroidal graph to obtain an optimal solution. Sloan et al. [24] locate bottlenecks and reduce the required searches. Cook et al. [6] use Fuchs' algorithm to minimize the surface area. Kehtarnavaz et al. [14] represent the search problem as a Levenshtein graph and use dynamic programming to find its minimum cost path. Furthermore, some fast heuristic methods have been developed for tiling. Strategy of Christiansen et al. [4] is on the shortest diagonal distance. Ganapathy et al. [10] use the concept of least tension as a heuristic guideline to tiling. Wang et al. [26] present another method: assigning an initial merit to each triangle and using relaxation to interactively refine these weights. The heuristic methods usually work well and fast on similar contours. These algorithms search for an optimal or a near optimal solution, but do not address the branching problem [15, 9, 24, 6, 14, 10, 26].

As to the dissimilar contour, such as in Fig. 2, the algorithms [15, 9, 24, 6, 14, 10, 26, 4, 7] which attempt to tile all contour vertices to the adjacent slice produce an unnatural topology as shown in Fig. 2(d). Boissonnat [2] and Barequet et al. [1] produce planar triangles which lie on the slice, to avoid the Fig 2(d) tiling. However the planar triangles distort the topology because of a lack of correlation in 3D. Zyda et al. [27] suggest some solutions including user interaction for different situations. Kehtarnavaz et al. [13] decompose contours into segments then blend similar contours into parametric surfaces. This approach works only on similar contours and generates smooth surfaces.

Some papers model the branching region by interpolating many intermediate contours [17] or by the homotopy model [23]. These methods generate a smooth surface but they either generate an excessive number of triangles or do not work on complicated branching. The rest of branching processing can be classified into four methods as shown in Fig. 3(b)-(e). Fig. 3(a) shows a contour $C3$ of $S2$ corresponding to two contours $C1$ and $C2$ of $S1$. Fig. 3(b) shows that a curve segment L is added between two slices to model the valley or saddle point formed by the branching. The added curve segment L is placed at contour $C3$ in (c). One or more line segments are added to form a composite contour as in (d); thereafter the tiling between the composite contour and $C3$ becomes one-to-one. Fig. 3(e) also forms a composite contour, which is the convex hull of branching contours, to have one-to-one tiling. The branching region between $C1$ and $C2$ is filled up by planar triangles. In terms of topological correctness, the best branching handling is (b) because it corresponds to the physical model better than the others do. Methods in Fig. 3(d) and (e) result in unnatural shading of the rendering surface since the composite contour has less 3D correlation with the actual physical model.

Christiansen et al. [4] and Shantz [22] use the method in Fig 3(d). They dip down the middle of the bridge to model the saddle point of the branching region. This approach does not work on complicated branching or a narrow curved branching region. Ekoule et al. [7] form an intermediate contour, similar to a composite contour, between two slices for the one-to-many branching. The intermediate contour is tiled to the merging contour as well as to the branching contours. As to the general many-to-many cases, their method assume that the Z direction is sampled adequately, so two corresponding contours have very close XY position. Barequet et al. [1] and Meyers et al. [19] use the scheme as in Fig 3(e). Meyers improves the planar triangle problem associated with this approach by feeding the triangulated mesh into a surface fitting program to regenerate the surface.

Boissonnat [2] invented a different approach other than tiling. He uses Delaunay triangulation on the contours. Then tetrahedra are formed between the corresponding contours. The surface of the polyhedron formed by the union of tetrahedra is the desired surface. He shows limitations in some topologies of his approach. Geiger [11] solves the related problems. He projects the external Voronoi skeleton (EVS) from one slice to the other and adds the projection in the Delaunay triangulation. Thus the triangles of a merging contour are split into several regions corresponding to each of the branching contours. Tetrahedra can be constructed between these corresponding regions. His method works well in complicated branching and in holes. His branching handling is as in Fig. 3(c). In the case of dissimilar contours such as in Fig. 2(b), his algorithm tiles the dissimilar area (i.e. the interior) of $C1$ to its EVS projection on the other slice. One problem, Geiger claims, is that contours having no neighbors in the adjacent slice will completely diminish.

From the analysis of previous work, we know that most of the surface based papers mentioned above somehow violate at least one of the following guidelines.

1. Concerning dissimilar contours, do not tile every contour vertex because it results in an unnatural topology.
2. The re-sampling of the reconstructed surface should be exactly as the original contours. The branching methods shown in Figure 4 (c), (d), and (e) violate this guideline.
3. Do not form composite contours since they do not correspond well to the actual physical model.

We present an algorithm which does not violate these guidelines. We present the theories behind our algorithm in Section 2, the implementation in Section 3, and our contributions in Section 4. We discuss our results and conclusion in Section 5. Lastly, the proofs of Section 2 are listed in the Appendix.

2 Corresponding and Tiling Rules

We define three criteria for the desired reconstructed surface ($\mathcal{SURF}$). The criteria are selected to let the reconstructed surfaces correspond well to the actual physical model. We limit these criteria between two adjacent slices. Please refer to the definitions and notations stated in this section.

Criterion 1 *$\mathcal{SURF}$ and the $\mathcal{PCR}$s (positive contour regions) form the piecewise closed surfaces of solid models.*

Criterion 2 *The data represented by the solid cannot flip signs more than once along any vertical line (a line parallel to Zaxis). So, any vertical line intersects $\mathcal{SURF}$ at no point, one point, or a line segment.*

Criterion 3 *The re-sampling of $\mathcal{SURF}$ on the slice should be exactly as the original contours.*

From these three criteria, we can derive precise tiling and corresponding rules. All theorems and lemmas are proved in Appendix. Theorem 1-5 are related to the tiling rules, and Theorem 6-7 describe the corresponding rules. From Theorems 6 and 7, the corresponding relationship is unique for the reconstructed surface to satisfy Criteria 1-3.

Definitions and notations:

1: Projection is onto the adjacent slice unless otherwise described. The notation of the projection of an object is denoted by appending a prime sign ($'$).

2: cross: intersect at exactly one point but not any end point.

3: A contour's direction is defined so the CT data of positive sign is on the left side of any contour segment. A positive contour has a CCW (counter clockwise) direction. For the example in Fig. 4(b), contour $C1$ is negative, and $C2$ is positive.

4: A vertex is an end point of a contour segment. It has the same sign as its contour.

5: $\mathcal{S}$ denotes the slice, and $\mathcal{I}(C)$ and $\mathcal{O}(C)$ denote the inside and outside regions of a simple polygon C respectively. $\mathcal{S} - (\mathcal{I}(C) \cup \mathcal{O}(C)) = C$.

6: $\mathcal{LS}(q)$ and $\mathcal{RS}(q)$ are the left side and the right side of a vertex q respectively.
Suppose $\overline{pq}$ and $\overline{qr}$ of Fig. 4(a) are two ordered contour segments. The two half lines $q\vec{p}$ and $q\vec{r}$ divide $\mathcal{S}$ into two regions. Then $\mathcal{LS}(q)$ is the shadow region which contains the left side of $\widehat{\overline{pqr}}$.
$\mathcal{RS}(q) = \mathcal{S} - (\mathcal{LS}(q) \cup q\vec{p} \cup q\vec{r})$

7: $\mathcal{NEC}$ is the nearest enclosing contour of a point or a contour. The slice boundary is considered as a negative contour. Every point or contour which does not intersect the slice boundary is enclosed by at least one contour. $\mathcal{NEC}$ is the nearest one. $C \subset \mathcal{I}(\mathcal{NEC}(C))$, here C could be a point or a contour. For the example in Fig. 4(b), $C1 = \mathcal{NEC}(V')$.

8: contour region and $\mathcal{PCR}$ (positive contour region): A contour region is the set of points which have a same $\mathcal{NEC}$. It has the same sign as the $\mathcal{NEC}$. The shadow region of Fig. 4(b) between $C1$ and $C2$ is one $\mathcal{PCR}$.

9: positive/negative $\mathcal{PV}$ (projection vertex) and $\mathcal{OV}$ (overlapping vertex): Let V be a vertex. (see Fig. 4(b)). If V' is on a contour, then V is an $\mathcal{OV}$. Suppose V is not an $\mathcal{OV}$. V is a positive or negative $\mathcal{PV}$ if V and $\mathcal{NEC}(V')$ have different or same signs respectively.

10: embedded contours: New vertices on contours are embedded to break contour segments, so that the intersections between the projections of contours are always contour vertices or contour segments. The reason of making embedded contours is stated in Lemma 3. All theorems apply to embedded contours.

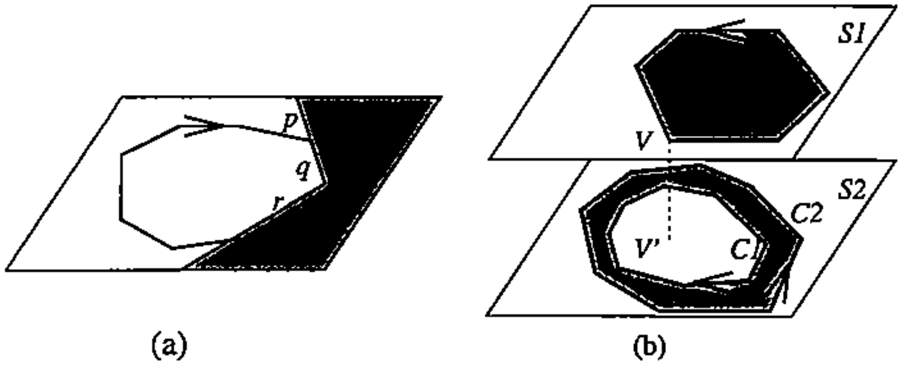


Figure 4: (a) The shadow region is $\mathcal{LS}(q)$. (b) Contours have directions. $C1 = \mathcal{NEC}(V')$.

11: $\mathcal{SURF}$: any reconstructed surface satisfying Criterion 1-3.

Lemma 1 *If a vertical line L not passing any contour intersects $\mathcal{SURF}$, then L passes exactly one $\mathcal{PCR}$.*

Lemma 2 *Let L be a vertical line not passing any contour. Suppose it has M ($M = 0, 1,$ or 2) intersections with the $\mathcal{PCR}$s on both slices, and it has N ($N = 0$ or 1) intersection with $\mathcal{SURF}$. Then,*

1. $M = 1 \Longleftrightarrow N = 1$
2. $M = 0$ *or* $M = 2 \Longleftrightarrow N = 0$

Lemma 3 *If the projections of two contour segments onto the XY plane cross each other, then they cannot be tiled.*

Because of Lemma 3, embedded contours (Definition 10) are formed to allow the tiling of crossed contour segments. From now on, all contours are embedded contours.

Theorem 1 *Any $\mathcal{OV}$ must tile to its projection.*

Theorem 2 *Let T be a tiling line segment incident with a contour vertex V.*

1. *Suppose V is not an $\mathcal{OV}$.*
   if $\mathcal{NEC}(V')$ is negative $\Longrightarrow T' \subset (\mathcal{S} - \mathcal{RS}(V))$, or
   if $\mathcal{NEC}(V')$ is positive $\Longrightarrow T' \subset (\mathcal{S} - \mathcal{LS}(V))$.
2. *If V is an $\mathcal{OV}$, its projection V' is also a vertex because of embedded contours. Then $T' \not\subset ((\mathcal{LS}(V) \cap \mathcal{LS}(V')) \cup (\mathcal{RS}(V) \cap \mathcal{RS}(V')))$.*

Theorem 3 *Let T be a tiling line segment, and C be any contour. Then T' cannot have intersections with both $\mathcal{I}(C)$ and $\mathcal{O}(C)$.*

Theorem 4 *Let T_2 be any existing tiling line segment, and T_1 be the proposed tiling line segment.*

1. *If T_1' crosses T_2', then T_1 cannot be a tiling line segment.*
2. *If T_1' intersects, but not crosses, T_2', and T_1 crosses T_2 (calculated in 3D), then T_1 cannot be a tiling line segment.*

Lemma 4 *Theorem 4.2 can be enforced by checking the intersection of T_1' and T_2' in 2D. Let () denote the open line segment. And all projections are onto the XY plane. Let the two end points of T_1 be u_1 and v_1, and let that of T_2 be u_2 and v_2.*

1. *If $T_1' \subset (u_2', v_2')$ or $T_2' \subset (u_1', v_1')$ as in Fig. 5(a), then T_1 crosses T_2.*
2. *If $(T_1' \cap T_2')$ is a line segment, then $\overrightarrow{u_1v_1}$ and $\overrightarrow{u_2v_2}$ either have a same direction or a reversed direction. Let u_1 and v_1 are renamed in a way such that $\overrightarrow{u_1v_1}$ and $\overrightarrow{u_2v_2}$ have a same direction as in Fig. 5(b). If u_1 and u_2 are on different slices, then T_1 crosses T_2.*

This lemma is observed from geometry examples and is not proved. Lemma 4.1 is still applicable to the case of vertical T_1 or T_2 (i.e. T_1' or T_2' is a point).

Lemma 5 *Any vertical line passing a tiling triangle (including 3 edges and inside) not violating Theorems 2 and 3 has an intersection with $\mathcal{SURF}$.*

Theorem 5 *Any tiling triangle not violating Theorems 1-4 can form part of $\mathcal{SURF}$.*

Lemma 6 *Let T be a tiling line segment incident with a vertex V on a contour C. If V is a positive $\mathcal{PV}$, then $T' \subset (\mathcal{I}(C) \cup C)$. If V is a negative $\mathcal{PV}$, then $T' \subset (\mathcal{O}(C) \cup C)$.*

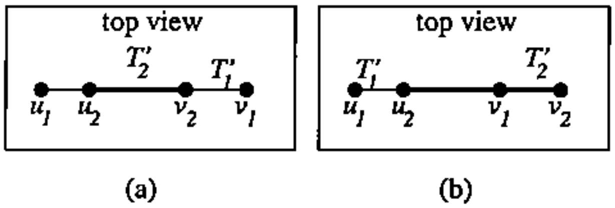


Figure 5: (a) $T_2' \subset T_1'$, (b) T_2' overlaps T_1' partly.

Theorem 6 *If a tiling triangle can be placed between two contours $C1$ and $C2$, then one of the following is true.*

1. *$C1'$ intersects $C2$.*
2. *If $C1$ and $C2'$ are disjoint, then they have different signs, each has at least one negative $\mathcal{PV}$, and their $\mathcal{NEC}$s do not insulate them (i.e. exists a non-$\mathcal{OV}$ vertex $V_1 \in C1$ such that $\mathcal{NEC}(V_1') = \mathcal{NEC}(C2)$, and vice versa.)*
3. *If one contour's ($C1$) projection is inside the other ($C2$), then they have the same sign, $C1$ has at least one negative $\mathcal{PV}$, $C2$ has at least one positive $\mathcal{PV}$, and there is no contour insulating $C1$ and $C2$ (i.e. exists a non-$\mathcal{OV}$ vertex $V_2 \in C2$ such that $\mathcal{NEC}(V_2') = \mathcal{NEC}(C1)$, and $C2 = \mathcal{NEC}(C1')$.)*

Theorem 7 *If any of the three conditions of Theorem 6 holds, then there exists a path on $\mathcal{SURF}$ linking these two contours. Note that this does not imply that a tiling triangle always exists in between.*

3 Implementation

There are different ways to implement an algorithm which follows the corresponding and tiling rules. Our implementation is just one example. It has the following major steps:

Step 1: form closed contours from image slices.

Step 2: make embedded contours (Definition 10).

Step 3: find correspondence between contours.

Step 4: form the tiling region of each vertex.

Step 5: form the optimal-tiling-vertex (OTV) table.

Step 6: do tiling.

Step 7: collect the boundary of un-tiled regions.

Step 8: form polygons to cover the un-tiled region based on its edge Voronoi diagram (EVD).

Our program can input image slices as well as contour data. In the case of contour data, Step 1 is skipped. The detailed implementation of each step is described in the following paragraphs.

The 2D marching cubes algorithm [18] is used to generate contour segments from an image slice. When ambiguity is encountered, a bi-linear function is fit to the grid to resolve the ambiguity. We assume that the CT volume objects do not intersect the slice boundary. If they do, we can set data on the slice boundary to be negative to form closed surfaces and later delete all reconstructed triangles connecting to the slice boundary to remove the pseudo surface. All generated contour segments can be linked to form simple polygons. Each contour segment from the 2D marching cubes algorithm is miniscule, so the contours are approximated by fewer contour segments under an error tolerance (see [8]). The approximation could result in the intersection of contours of the same slice if a gross tolerance is used. We usually choose half pixel as the approximation tolerance which effectively eliminates 75%-90% of contour segments and rarely causes the intersection.

Based on Lemma 3, the embedded contours are formed. Theorem 6 judges the correspondence between contours on different slices. One table for each slice stores the contour relationships (disjointedness or enclosure) of the same slices. Another table stores the contour relationships (intersection, disjointedness, or enclosure) between adjacent slices. The outermost contour is considered to be positive. Each contour has an opposite sign to its $\mathcal{NEC}$. The sequence of contour segments might be reversed to make a positive contour take a CCW direction. The $\mathcal{NEC}$ of each vertex projection is recorded. Once this information is available, Theorem 6 can determine the correspondence between contours.

The tiling region of each vertex is defined in Theorem 2. Two line equations and an AND-OR flag define the tiling region of a vertex described by Theorem 2.1. If V is an $\mathcal{OV}$, then V' is also an $\mathcal{OV}$, and V' is the OTV (optimal tiling vertex) of V. So, the tiling region of V' can be indexed from the OTV table, and verifying Theorem 2.2 does not need extra data structure.

The OTV table stores the OTVs of all vertices. The OTV

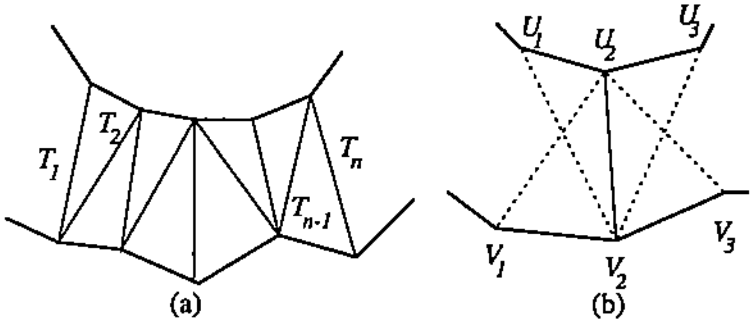


Figure 6: (a) Only the boundary tiling segments (T_1 and T_n) of a tiled region need to be checked for Theorem 4. (b) the selection of tiling triangle is based on the minimum tiling distance.

on the adjacent slice is searched for each vertex V. Every vertex on a corresponding contour is tested to tile to V. The tiling must satisfy Theorems 2 and 3. The eligible vertex having the shortest distances to V is selected as the OTV of V. The current implementation sorts the candidate vertices based on their distances to V. Then the candidate vertices, starting from the closest vertex, are checked until one satisfies Theorems 2 and 3. Usually, the closest candidate is qualified. So the average time complexity is $O(n^2 \log n)$. But in the worst case of no qualified vertex, every vertex on the corresponding contour is tested. So this implementation has an $O(n^3)$ worst case time complexity. (Checking Theorem 3 needs $O(n)$ time and there are $O(n)$ checking for each V in the worst case.) Our method can be done much faster through the following. All vertices can be presorted into an RPO tree [5]. So the closest-point query can be done in $O(\log n)$ time. Furthermore, the $O(\log n)$ algorithm described by Preparata [21] can be used to check Theorem 2. The time complexity of this approach is $O(n \log n)$ average case and $O(n^2 \log n)$ worst case. Stopping the search after a certain range will lower the leading time constant of the worst case time complexity.

During the tiling, it is often to check whether the tiling segment is legal (satisfying Theorems 2-4) or not. Verifying Theorem 2 for a proposed tiling line segment takes $O(1)$ time. Theorem 3 takes $O(n)$ or $O(\log n)$ time depending on implementation. Verifying Theorem 4 takes $O(n)$ time because the number of existing tiling line segments is always increasing and can not be pre-processed. Fortunately, it is not necessary to check the intersection of the proposed tiling segment with all existing tiling segments. Only the boundary tiling segments of tiled regions need to be verified. For the example in Fig. 6(a), only the two boundary tiling segments T_1 and T_n are needed to be checked. It is impossible for the proposed tiling segment to cross any of T_2 to T_{n-1} without crossing T_1, T_n, or any contour segment. Furthermore, only the existing tiling segments, which have one or two end points on the same contour as the proposed tiling segment, need to be verified.

There are four passes in tiling. The tiling sequence is based on the optimality of the tiling pair. Suppose two vertices U_1 and V_1 are on the corresponding contours, and $\overline{U_1V_1}$ is legal. They can be classified into four cases in the order of optimality. The first pass only tiles Case 1, and so on.

Case 1: U_1 is the OTV of V_1, and vice versa.

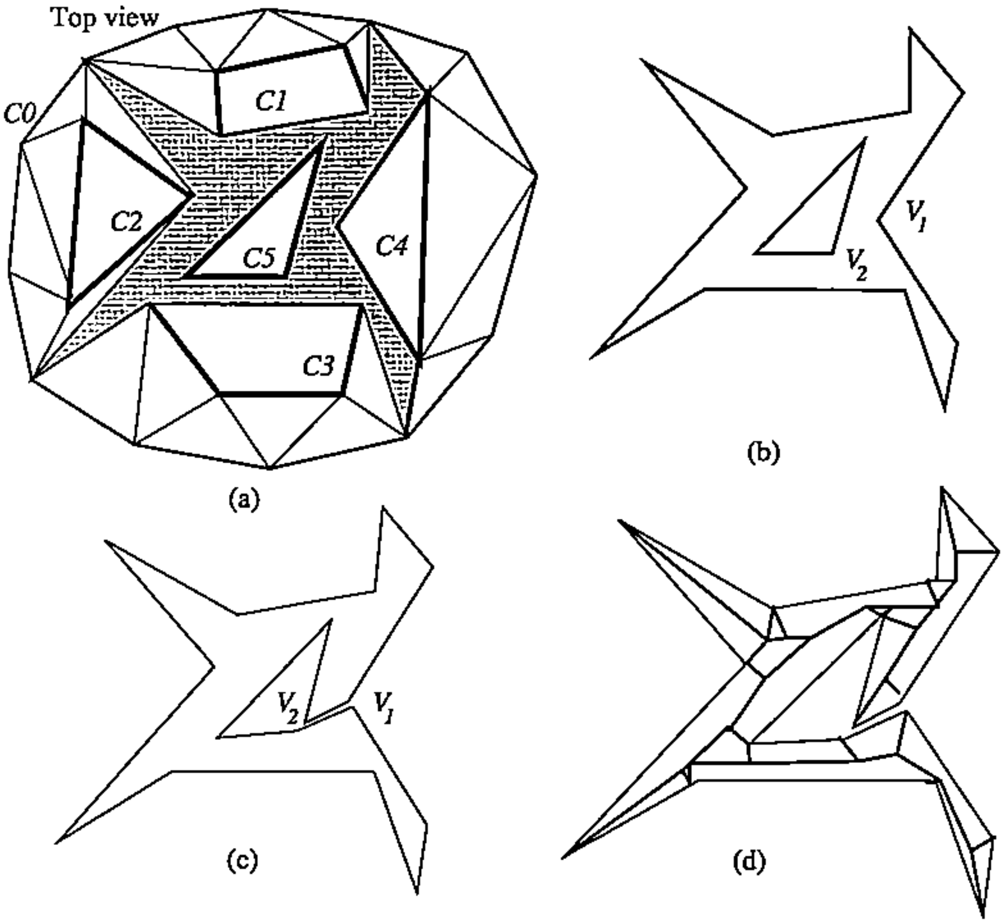
Case 2: One vertex (U_1) is the OTV of the other (V_1), but not vice versa, and V_1 and the OTV of U_1 are on the same contour.

Case 3: One and the OTV of the other are on the same contour, and vice versa.

Case 4: the rest cases.

In the first pass, the OTV table is scanned to find all optimal tiling pair. This is an $O(n)$ time for finding all pairs. Two tiling triangles on both sides of the tiling segment are formed based on the shortest tiling distance metric. Suppose U_2 and V_2 is the tiling pair, there are four possible tiling triangles sharing $\overline{U_2V_2}$ as shown in Fig. 6(b). One legal tiling triangle is chosen for each side of $\overline{U_2V_2}$. The boundary tiling line segments and their directions (spanning right or left) are put into a boundary array. If the to-be-stored boundary tiling segment is already in the array, it is already shared by two tiling triangles, and thus it is deleted from the boundary array.

The Cases 2-4 are done in the other three passes. The starting tiling pairs are popped out from the boundary array, and the tiling spans on one direction until no satisfying case is available. The starting tiling pairs of the second pass also come from the scanning of the OTV table. Like in the first pass, the boundary tiling segments are stored in an alternative boundary array. The tiling takes $O(n^2)$ time because checking Theorem 4 takes $O(n)$ time, and there are $O(n)$ proposed tiling segments. Compared to making only one pass to do all tiling, our multi-pass approach does not increase the number of proposed tiling pairs, so it does not significantly increase



(a)

(b)

(c)

(d)

Figure 7: Covering the un-tiled region: (a) Contours $C1$-$C5$ of the top slice correspond to $C0$ of the bottom slice. (b) the un-tiled region. (c) collapsing two polygons into one simple polygon by adding $\overline{V_1V_2}$. (d) The un-tiled regions is covered by polygons formed by the region boundary and the EVD (edge Voronoi diagram). The Z value of EVD is set to the middle of two slices.

the processing time.

After all tiling is done, there could be some un-tiled regions in dissimilar contours, holes or branching regions. The processing of un-tiled region can be illustrated in Fig. 7. Contours $C1$-$C5$ of the top slice correspond to $C0$ of the bottom slice. The un-tiled regions can be traced from the un-used contour segments and the boundary tiling segment array. In some rare cases, the projection of an un-tiled region could be a non-simple polygon. In such a case, the projection is broken into simple polygons. In the case of one un-tiled region's projection enclosing the projection of another un-tiled region as in Fig. 7(b), these two polygon projections are collapsed into one simple polygon by adding $\overline{V_1V_2}$ which is the shortest vertex pair. We use Lee's algorithm [16] to find the EVD (edge Voronoi diagram or medial axis). Because of its divide-and-conquer nature, the contour vertices can be represented as the leaves of a tree, and all vertices on the EVD are non-leaf nodes. Non-leaf nodes are merged on the condition that the polygons formed by their descendents do not cross each other. So fewer triangles are required to cover the un-tiled region. Then the polygon covering the un-tiled region is traced by walking from a leaf node up and down to its neighboring leaf node. Finally polygons are triangulated. The Z values of the EVD and the middle point of $\overline{V_1V_2}$ are set to the middle of two slices.

4 Contributions

Our research offers two major contributions. The first is the development of a theoretical backbone for our algorithm. We define three criteria for the desired surfaces and derive precise rules to achieve these criteria. This has not been done previously. As discussed in Section 1, other papers using heuristic methods for checking correspondence have no guaranteed success in different topologies. Our algorithm can generate the desired surfaces from any topology. We do not claim that it generates the same topology as that of the actual physical model if the slice distance is great. In such an unconstrained case, no algorithm guarantees the generation of the actual topology. Our algorithm, given any input data, generates a unique corresponding topology, which satisfies the desired surface criteria.

The second major contribution is the development of a robust algorithm whose reconstructed surfaces correspond well to the actual model. This results in a natural looking reconstructed surface rendering. As discussed in Section 2, the data of our reconstructed model along any vertical line cannot flip its sign more than once between two slices. The re-sampling of the reconstructed surfaces is exactly as the original contours. And our branching method corresponds to Fig. 3(b) which is better than Fig. 3(c)-(e). All these factors together make our algorithm correspond well to the actual physical model.

A less major, but still significant, contribution of our research is our new tiling algorithm. Traditional tiling algorithms have problems in the branching region because it is difficult to know when the tiling should be alternated to other branching contours. Many papers reduce the tiling into one-to-one using composite contours or intermediate contours. As discussed in Section 1, this approach has drawbacks. Our approach does not form any composite contour or intermediate contour. It makes several passes in which the optimal pairs are tiled in the first pass, and so on. This algorithm has a near optimal tiling result even in the branching regions because the tiling is done in the order of its optimality.

5 Results

Two sets of results from actual medical data are presented in this section. Fig. 8 shows the Gouraud shading and the wire frame of the reconstructed surface of a brain hemisphere. It is generated from a set of contour data which is manually traced from 52 MRI image slices. The other picture, Fig. 9(a), shows a reconstructed skull. The noise around the teeth is inherent in the original image slices. The surface is automatically generated from 112 256*256 CT slices. The image slices are pre-processed by a 3*3 median filter. Fig. 9(b)-(e) shows two adjacent slices around the nasal and the tiling in between. As can be seen from (b) and (c), there are numerous holes, dissimilar areas, and branching between two adjacent slices. Therefore, these two examples show the capability of our algorithm to handle complicated topologies. The marching cubes approach generates 554,500 triangles from the skull data set. The tolerance error during the approximation of the skull contours is 0.5 pixel. Our approach generates a significantly lower number (54,071) of triangles compared to the marching cubes approach. Table 1 summaries our results. The CPU time is based on SUN Sparc IPC workstations.

6 Conclusion

We have presented a robust algorithm in reconstructing surfaces from a set of contour data or image slices. The theoretical derivation of the corresponding and tiling rules allow our algorithm, given any input data, to generate a unique topology satisfying the desired surface criteria. The reconstructed surface corresponds well to the actual physical model and produces the appearance of a natural surface rendering. It generates a significantly lower number of triangles compared

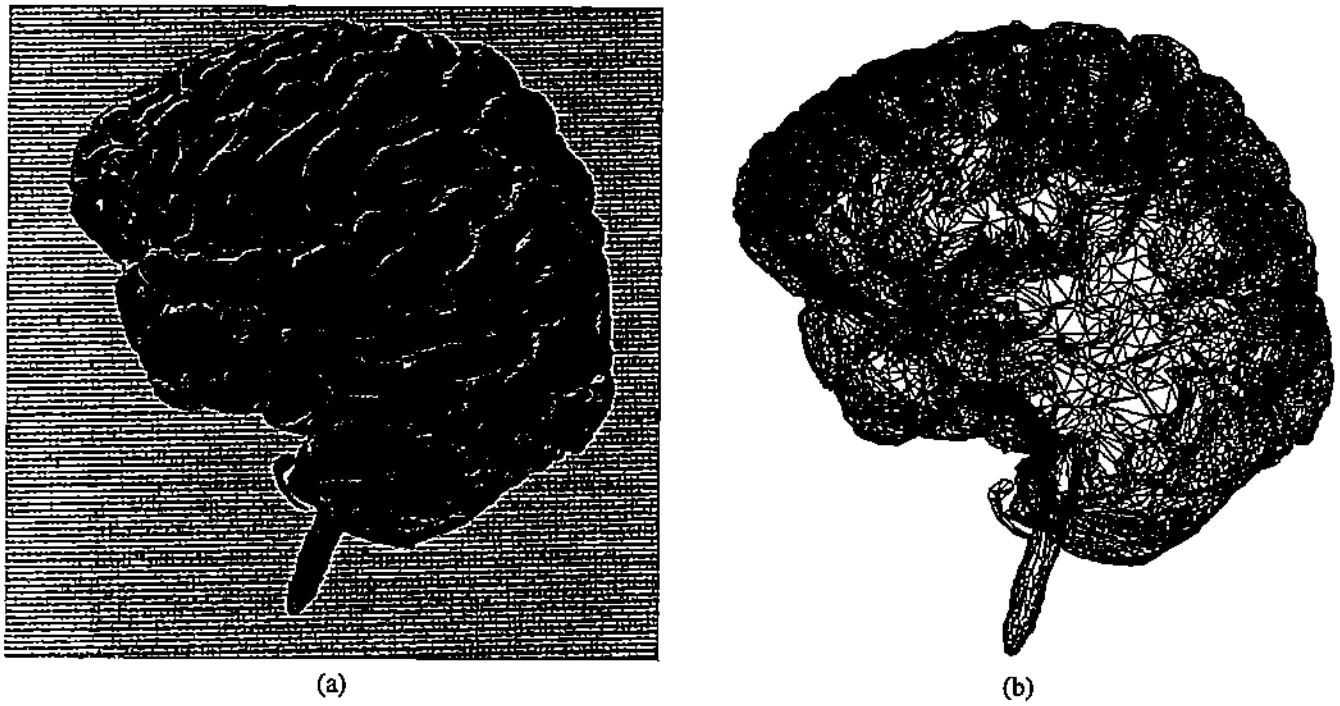
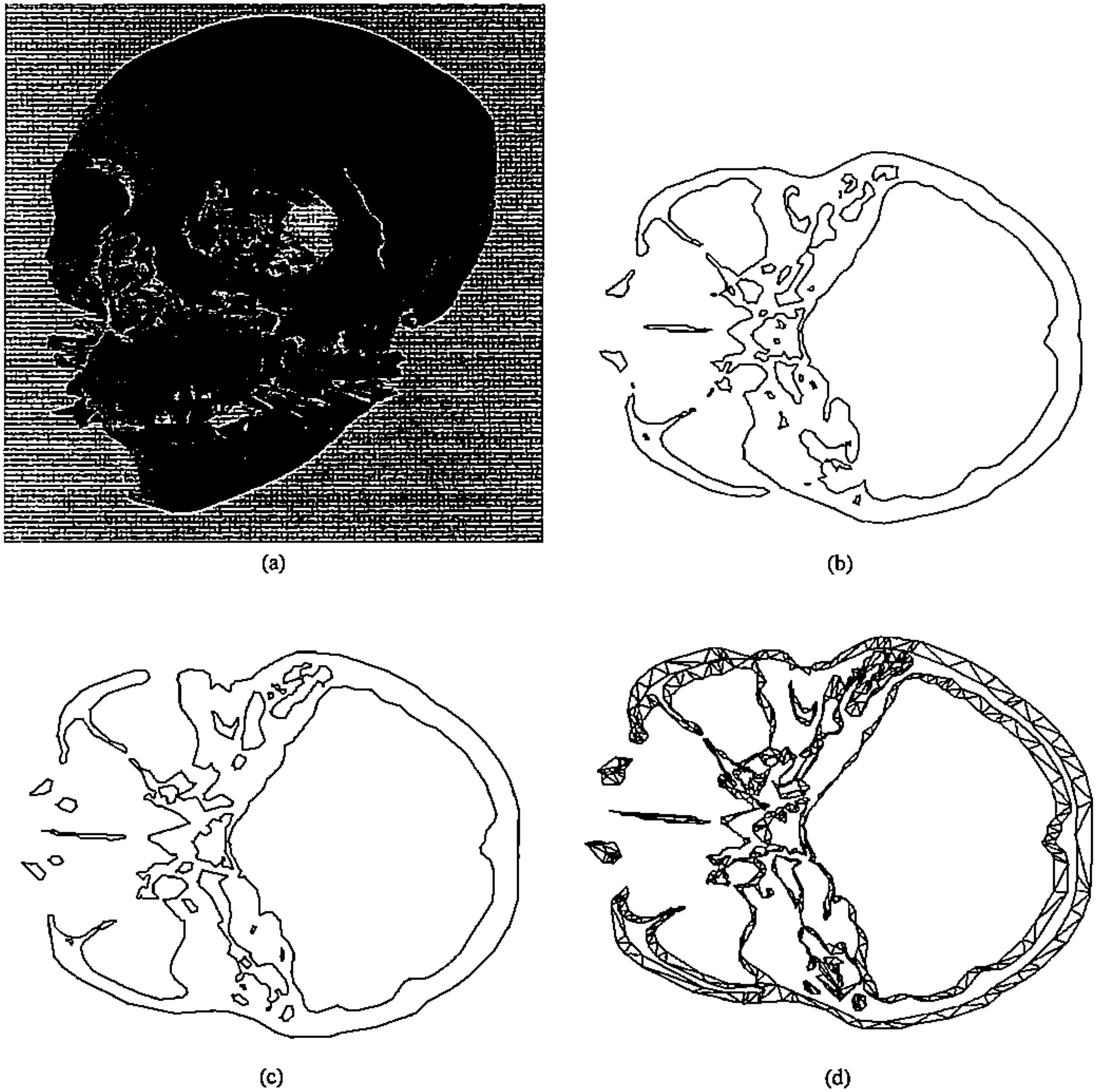


Figure 8: Visualization of a reconstructed brain hemisphere (a) Gouraud shading, (b) wire frame

	segmentation method	# of slices	# of Δ	CPU time (sec)	marching cubes Δ	resolution	approximation error
brain	manually	52	37,992	779	N/A	N/A	N/A
skull	auto.	112	54,071	1,859	554,500	256*256	0.5 pixel

Table 1: Results



(a)

(b)

(c)

(d)

Figure 9: Visualization of a reconstructed skull: (a) Gouraud shading, (b) & (c) two sample slices, (d) the tiling of (b) & (c).

to the marching cubes approach. The processing time of our current implementation can be shortened by the methods discussed in Section 4. The results of actual medical data are presented to show the capability of our algorithm. With these features, our algorithm is valuable in surface reconstruction and visualization.

Acknowledgements

We are grateful to the University of North Carolina for anonymous ftp access to the skull and brain data sets. Our thanks also to Dan Schikore for his development of visualization tools. This research was supported in part by NSF grants CCR 92-22467 and AFOSR grants F49620-93-10138 and F49620-94-10080, and ONR grant N00014-91-1-0370.

References

[1] G. Barequet and M. Shair. Piecewise-linear interpolation between polygonal slices. In *10th Computational Geometry*, pages 93–102, 1994.

[2] J. D. Boissonnat. Shape reconstruction from planar cross sections. *Comp. Vis. Grap. & Image Proc.*, (44):1–29, 1988.

[3] Y. Bresler, J. A. Fessler, and A. Macovski. A bayesian approach to reconstruction from incomplete projections of a multiple object 3d domain. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 11(8):840–858, Aug. 1989.

[4] H. N. Christiansen and T. W. Sederberg. Conversion of complex contour line definitions. *Computer Grapics*, pages 187–192, Aug. 1978.

[5] K. L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, (4):830–847, 1988.

[6] L. T. Cook, P. N. Cook, K. R. Lee, S. Batnitzky, B. Wong, S. L. Fritz, J. Ophir, S. J. Dwyer III, L. R. Bigongiari, and A. W. Templeton. An algorithm for volume estimation based on polyhedral approximation. *IEEE Trans. Biomed. Eng.*, 10(6):949–955, Nov. 1988.

[7] A. B. Ekoule, F. C. Peyrin, and C. L. Odet. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. Graphics*, 10(2):182–199, Apr. 1991.

[8] David Eu and Godfried T. Toussaint. On approximating polygonal curves in two and three dimensions. *Comp. Vis. Grap. & Image Proc.*, (3):231–246, 1994.

[9] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from plannar contours. 20:693–702, Oct. 1977.

[10] S. Ganapathy and T. G. Dennehy. A new general triangulation method for planar contours. *Computer Grapics*, pages 69–75, 1982.

[11] B. Geiger. *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*. PhD thesis, INRIA, France, 1993.

[12] C. Gitlin, J. O'Rourke, and V. Subramanian. On reconstructing polyhedra from parallel slices. Technical Report 25, Dept. Comput. Sci., Smith College, Northampton, MA, 1993.

[13] N. Kehtarnavaz and R.J.P. De Figueiredo. A framework for surface reconstruction from 3d contours. *Comp. Vis. Grap. & Image Proc.*, (42):32–47, 1988.

[14] N. Kehtarnavaz, L. R. Simar, and R.J.P. De Figueiredo. A syntactic/semantic technique for surface reconstruction from cross-section contours. *Comp. Vis. Grap. & Image Proc.*, (42):399–409, 1988.

[15] E. Keppel. Approximating complex surface by triangulation of contour lines. *IBM Journal of Research & Development*, (19):2–11, 1975.

[16] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. on Patt. Anal. Mach. Intell.*, PAMI-4(4):363–369, Jul. 1982.

[17] W. C. Lin and S. Y. Chen. A new surface interpolation technique for reconstructing 3d objects from serial cross-sections. *Comp. Vis. Grap. & Image Proc.*, pages 124–143, 1989.

[18] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface reconstruction algorithm. *Computer Grapics*, pages 163–169, 1987.

[19] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. Graphics*, 11(3):228–258, Jul. 1992.

[20] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara, and M. J. Wozny. Geometrically deformed models: A method for extracting closed geometric models from volume data. *Computer Grapics*, 25(4):217–226, Jul. 1991.

[21] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[22] M. Shantz. Surface definition for branching contour defined objects. *Computer Grapics*, pages 242–270, Jul. 1981.

[23] Y. Shinagawa and T. L. Kunii. The homotopy model: a generalized model for smooth surface generation from cross sectional data. *The Visual Computer*, pages 72–86, 1991.

[24] K. R. Sloan and J. Painter. Pessimal guesses may be optimal: a counterintuitive search result. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 10(6):949–955, Nov. 1988.

[25] B. I. Soroka. Generalized cones from serial sections. (15):154–166, 1981.

[26] Y. F. Wang and J. K. Aggarawal. Surface reconstruction and representation of 3-d scenes. *Pattern Recognition*, 19(3):197–207, 1986.

[27] M. J. Zyda, A. R. Jones, and P. G. Hogan. Surface construction from planar contours. *Computer & Graphics*, 11(4):93–408, 1987.

Appendix

In addition to the definitions in Section 2, we further define $\mathcal{DISK}(P)$ as the disk region on slice centered at a point P. The radius is arbitrary small.

proof of Lemma 1:
Suppose L does not pass any $\mathcal{PCR}$ or contour, and it intersects $\mathcal{SURF}$. Base on Criterion 2, the intersection must be exactly one line segment or one point. The signs of the solid data along L are negative at both ends and are neutral at one line segment or one point. That means $\mathcal{SURF}$ needs to curve back at L. Then there exists a vertical line $L1$ very close to L such that $L1$ goes through $\mathcal{SURF}$ twice. This violates Criterion 2. So L must intersect at least one $\mathcal{PCR}$. Suppose L intersects two $\mathcal{PCR}$s. We can apply the same argument to the complement (dual) of the solid to contradict this assumption. So L must pass exactly one $\mathcal{PCR}$. □

proof of Lemma 2:
$N = 1 \Longrightarrow M = 1$ (Lemma 1). If $M = 1$, then the solid data at the intersection end of L is positive, and that of the other end is negative. It implies that there is a zero crossing (intersection) in between. So $M = 1 \Longrightarrow N = 1$, thus $M = 1 \Longleftrightarrow N = 1$. Lemma 2.2 is proved by inverting both sides of Lemma 2.1. □

proof of Lemma 3:
As shown in Fig. 10(a), c_1 and c_2 are two contour segments on different slices. Suppose ${c_1}'$ crosses c_2 at P', then $\mathcal{DISK}(P')$ is divided into R_1, R_2, R_3 and R_4. The $\mathcal{PCR}$s are shown as the shadow regions. Based on Lemma 2, the projection of the $\mathcal{SURF}$ is in $R_1 \cup R_3$, and not in $R_2 \cup R_4$. But the projection of the reconstructed triangle containing c_2 should either in $R_1 \cup R_2$ or in $R_3 \cup R_4$. Similar argument applies to c_1. So c_1 and c_2 cannot be tiled. □

proof of Theorem 1:
Let V be an $\mathcal{OV}$. A vertical line passing V have two point intersections with the contours. In order to satisfy Criterion 2, the intersection with $\mathcal{SURF}$ must be one line segment containing V and V'. V' is also an vertex due to embedded contours. So V tiles to V'. □

proof of Theorem 2

1. Suppose $\mathcal{NEC}(V')$ is negative as shown in Fig. 4(c). Then $\mathcal{DISK}(V')$ does not intersect any $\mathcal{PCR}$. There exists a point $V_1 \in (\mathcal{DISK}(V) \cap \mathcal{RS}(V))$ such that V_1 and ${V_1}'$ are not on any contour. Let L be the vertical line passing V_1. L does not intersect any $\mathcal{PCR}$ because the $\mathcal{PCR}$ incident with V is on $\mathcal{LS}(V)$ (see Definition 2.3). So, L does not intersect $\mathcal{SURF}$ according to Lemma 2. It implies $T' \not\subset \mathcal{RS}(V)$. So $T' \subset (\mathcal{S} - \mathcal{RS}(V))$. The case of positive $\mathcal{NEC}(V')$ is proved in the similar way.

2. If $T' \subset ((\mathcal{LS}(V) \cap \mathcal{LS}(V')) \cup (\mathcal{RS}(V) \cap \mathcal{RS}(V'))$. there exist a vertical line L passing a non-vertex point $Q \in (T' \cap \mathcal{DISK}(V))$ such that L does not pass any contour. L has two or zero intersections with $\mathcal{PCR}$s on both slices, and one intersection with T. This contradicts Lemma 2. □

proof of Theorem 3:
It is proved by contradiction. Suppose $T' \cap \mathcal{I}(C) \neq \phi$ and $T' \cap \mathcal{O}(C) \neq \phi$. T' has three ordered sections t_1, t_2 and t_3 as shown in Fig. 10(b) such that $t_1 \subset \mathcal{I}(C)$, $t_2 \subset C$, $t_3 \subset \mathcal{O}(C)$, and $t_1 \cup t_2 \cup t_3$ is a line segment. t_2 could be a line segment or a point. Both t_1 and t_3 are line segments. Suppose C is on slice $S1$, and the adjacent slice is denoted $S2$. Let V_1 and V_2 be the two end points of t_2, and V_1 is closer to t_1. (In the case that t_2 is a point, V_1 and V_2 are the same point.) There can not be any $\mathcal{OV}$ on t_2, otherwise T intersects the vertical tiling line segment incident with this $\mathcal{OV}$ (Theorem 1), and thus violates the solid surface. So $\mathcal{DISK}({V_1}')$ does not intersect any contour on $S2$. A vertical line $L1$ passing $t_1 \cap \mathcal{DISK}(V_1)$ does not pass any contour. $L1$ intersects T. So $L1$ must pass exactly one $\mathcal{PCR}$ based on Lemma 2. Same reasoning applies to the vertical line $L2$ passing $t_3 \cap \mathcal{DISK}(V_2)$. Because either $L1$ or $L2$ passes a $\mathcal{PCR}$ on $S1$, either $L2$ or $L1$ passes a $\mathcal{PCR}$ on $S2$. So there must be a contour on $S2$ crossing t_2', and form at least one $\mathcal{OV}$ on t_2. This contradicts the reasoning that t_2 should have no $\mathcal{OV}$. □

proof of Theorem 4:

1. Let Δ_1 and Δ_2 be the tiling triangles containing T_1 and T_2 respectively. $T_1' \cap T_2' = P'$. There are three cases to consider as shown in Fig. 11: 1) both Δ_1 and Δ_2 are non-vertical, 2) one is non-vertical, and 3) both are vertical. According to Criterion 2, $\mathcal{SURF}$ is single sheeted except at vertical triangles. So case 1 is invalid. Suppose Δ_1 is vertical and Δ_2 is non-vertical. Let surface $\mathcal{F}_1$ consist of all vertical triangles whose projections overlap with T_1'. $\mathcal{F}_1'$ intersects Δ_2' at a line segment. Let Q' be the other intersection of T_1' and Δ_2'. Because of Criterion 2, any vertical line passing $\overline{P'Q'}$ can not have two intersections with $\mathcal{SURF}$. So, the intersection of Δ_2 and $\mathcal{F}_1$ in 3D is a line segment whose projection is $\overline{P'Q'}$. Thus we can form at least three new triangles (two on Δ_2 and one on $\mathcal{F}_1$) sharing that line segment. This violates the solid surface. If Δ_2 is also vertical. Suppose surface $\mathcal{F}_2$ consists all vertical triangles whose projections overlap with T_2. $\mathcal{F}_1'$ crosses $\mathcal{F}_2'$. So the intersection of $\mathcal{F}_1$ and $\mathcal{F}_2$ could be a vertical line or a point P. If it is a vertical line segment, we can form four new triangles sharing that line segment and hence violates the solid surface. Suppose the intersection is a point P. There exists a

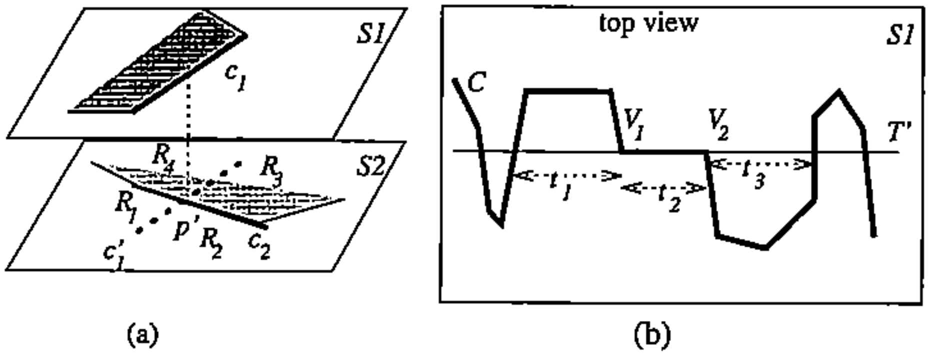


Figure 10: (a) used in the proof of Theorem 3, (b) crossed contour segments cannot be tiled so they need to be embedded with new vertex at P'.

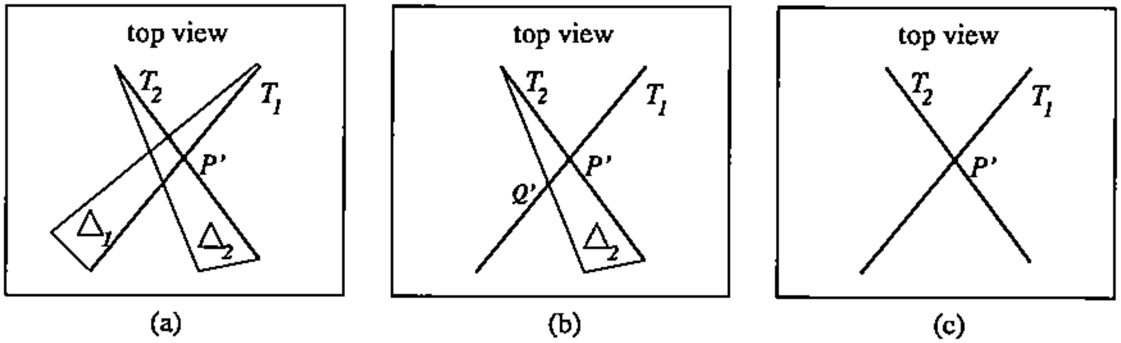


Figure 11: (a) Both triangles containing T_1 and T_2 are non-vertical, (b) One triangle is vertical. (c) Both triangles are vertical.

vertical line L passing $\mathcal{DISK}(P)$, and L does not pass any contour such that L intersect $\mathcal{SURF}$ twice. (The intersections are with the parts of $\mathcal{SURF}$ connecting to $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively.) This violates Criterion 2.

2. If T_1 crosses T_2 (in 3D), their point intersection is not a vertex. Then the tiling triangles containing T_1 and T_2 intersect at a non-vertex. Thus it violates the solid surface. □

proof of Lemma 5:
Suppose $\triangle ABC$ is a tiling triangle as shown in Fig. 12(a) and (b). $\overline{BC}$ is a contour segment. $\overline{AB}$ and $\overline{AC}$ do not violate Theorems 2-3. Let L be the vertical line passing any point $P \in \triangle ABC$. $\overline{A'B}$ or $\overline{A'C}$ can not intersect both the inside and outside of any contour (Theorem 3). So P' is either on contour or on the same side of any contour for any $P \in \triangle ABC$. If L does not pass any contour segment, L has the same number of intersection with $\mathcal{SURF}$ given any $P \in \triangle ABC$. If $\mathcal{NEC}(A')$ is negative, then $\overline{AB'}$ and $\overline{AC'}$ is in $(\mathcal{S} - \mathcal{RS}(A))$ (Theorem 2). So L has one intersection with the $\mathcal{PCR}$ on $\mathcal{LS}(A)$, and it has no intersection with any $\mathcal{PCR}$ on $S2$ because of negative $\mathcal{NEC}(A')$. So L intersects $\mathcal{SURF}$ according to Lemma 2. □

proof of Theorem 5:
Suppose a $\mathcal{SURF}$ exist. It contains existing tiling triangles and other triangles filling the gap between these tiling triangles. If we can construct a new $\mathcal{SURF}$ containing the proposed tiling triangle and the existing tiling triangles, then this theorem is proved.

Suppose $\triangle ABC$ in Fig. 12(a) is the proposed tiling triangle. $\overline{AB}$ and $\overline{AC}$ do not violate Theorem 1-4. $A \in S1$ and $B, C \in S2$. There are two cases: 1) $\triangle ABC$ is non-vertical, 2) $\triangle ABC$ is vertical.

If $\triangle ABC$ is non vertical, $\triangle A'BC$ does not degenerate into a line segment. From Lemma 5, every point in $\triangle ABC$ has a projection on $\mathcal{SURF}$. Let $\mathcal{F}_1$ be the set of all points on $\mathcal{SURF}$ whose projection is on $\triangle A'BC$. We can construct a new surface $M = (\mathcal{SURF} - \mathcal{F}_1) \cup \triangle ABC$. Because of Theorem 4, $\mathcal{F}_1$ does not contain any non-zero area part of existing tiling triangle. (It could contain edges or vertices of tiling triangles.) This means that the M does not affect the existing tiling triangle. The M might have vertical gaps along $\overline{AB}$ or $\overline{AC}$ as shown in the gaps between the shaded regions and $\triangle ABC$. However, these gaps can be filled by adding vertical triangles, so the new surface can satisfy Criterion 1-3.

If $\triangle ABC$ is vertical, $\triangle A'BC$ degenerates into a line segment. A' is not on the open line segment (B, C) otherwise there is an $\mathcal{OV}$ on (B, C), and $\triangle ABC$ is not a tiling triangle because $\overline{BC}$ contains more than one contour segment. Let the naming of vertices be as in Fig. 12(b).

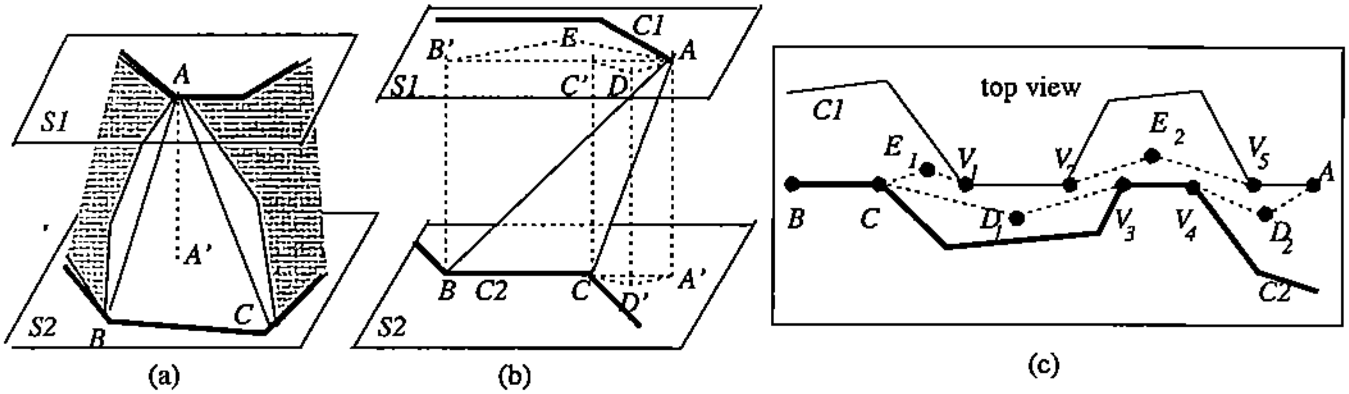


Figure 12: used in the proof of Theorem 5

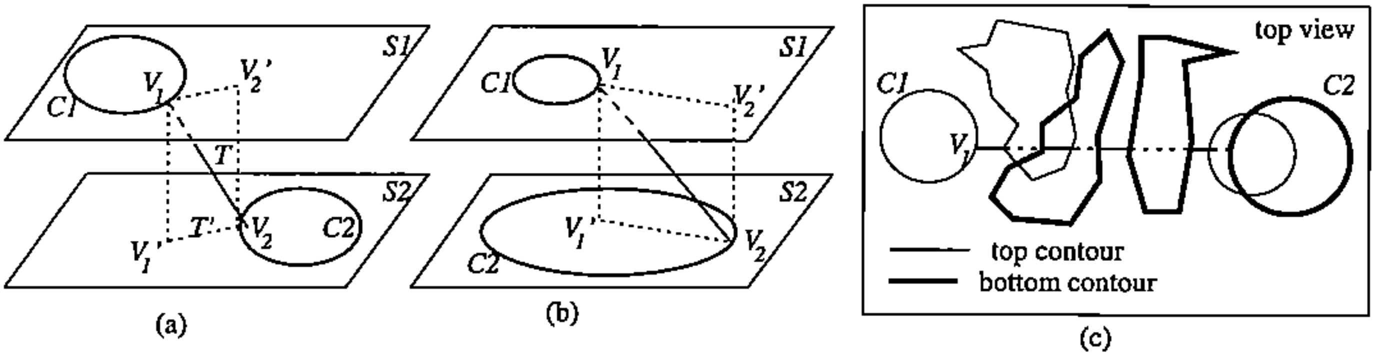


Figure 13: used in the proof of Theorems 6-7, (a) $C1$ and $C2$ are disjoint, (b) $C1'$ is enclosed by $C2$, (c) there exist a path on $\mathcal{SURF}$ linking $C1$ and $C2$,

We first prove the case that no contour segment other than $\overline{BC}$ is on $\overline{AB'}$ or $\overline{A'B}$. As shown in Fig. 12(b), we pick up a point E which is very close to $\overline{AB'}$ and the vertical line passing E intersects $\mathcal{SURF}$. A point D is picked by the same method to be close to $\overline{A'C}$. Let $\mathcal{F}_1$ is the set of all points on $\mathcal{SURF}$ whose projection is on and inside the simple polygon $AD'C'B'E$. $\mathcal{F}_1$ does not contain any non-zero area part of any other tiling triangle because no contour segments other than $\overline{BC}$ is on $\overline{AB'}$. We can construct a new surface $M = (\mathcal{SURF} - \mathcal{F}_1) \cup \triangle ABC$. The reason to have a simple polygon of non-zero thickness is to ensure that $\mathcal{SURF} - \mathcal{F}_1$ does not intersect $\triangle ABC$ at all. So the gaps in M can be filled by non-vertical triangles to make M satisfying Criterion 1-3. $\mathcal{F}_1$ does not contain any non-zero area part of existing tiling triangle. So this case is proved.

Suppose there are contour segments other than $\overline{BC}$ on $\overline{AB'}$ or on $\overline{A'B}$. We use the same method to add line segments as shown in Fig. 12(c). These line segment and parts of $\overline{AB'}$ can form only one simple polygon $AD_2V_4V_3D_1CE_1V_1V_2E_2V_5$ because of no $\mathcal{OV}$ on $\overline{AB'}$. Let $\mathcal{F}_1$ be the set of all points on $\mathcal{SURF}$ whose projection is onto the simple polygon $AD_2V_4V_3D_1CE_1V_1V_2E_2V_5$. In this case, $\mathcal{F}_1$ could contain non-zero area part of other tiling triangles; let one of them is named $\triangle_2$. The contour segment of $\triangle_2$ is on $\overline{A'B}$. $\triangle_2$ must be vertical so that its tiling line segments do not cross $\overline{AC}$. They are vertical, and their projections lie on $\overline{A'B}$. $\triangle_2$ does not intersect $\triangle ABC$ at non-zero area part because of Theorem 4. So we can put $\triangle_2 \cap \mathcal{F}_1$ back to M to not affect any existing tiling triangle. The gap between $\triangle_2$ and $\triangle ABC$ can be filled with vertical triangles. As in the previous case, the other gaps whose projection is not on $\overline{AB}$ can be filled by non-vertical triangle. Thus M can be made to be a new $\mathcal{SURF}$. □

proof of Lemma 6:

From Definition 9, there are two cases to form a positive $\mathcal{PV}$ V.

case 1: V is positive, and $\mathcal{NEC}(V')$ is negative.

case 2: V is negative, and $\mathcal{NEC}(V')$ is positive.

In case 1, $T' \subset (\mathcal{S} - \mathcal{RS}(V))$ according to Theorem 2. $T' \subset (\mathcal{I}(C) \cup C)$ because the $\mathcal{RS}$ of a positive vertex is outside the contour.

In case 2, $T' \subset (\mathcal{S} - \mathcal{LS}(P))$ according to Theorem 2. $T' \subset (\mathcal{I}(C) \cup C)$ because the $\mathcal{LS}$ of a negative vertex is outside the contour.

The second part of this lemma is proved similarly. □

proof of Theorem 6: Condition 1 is obvious based on Theorem 1.

Suppose $C1$ and $C2$ are disjoint. Let T be a legal tiling line segments incident with V_1 of $C1$ and V_2 of $C2$ as shown in Fig 13(a). $T' \subset (\mathcal{O}(C1) \cup C1)$, and $T' \subset (\mathcal{O}(C2) \cup C2)$. So V_1 and V_2 are negative $\mathcal{PV}$ (Lemma 6). There is no contour passing the open line segments (V_1, V_2') or (V_1', V_2) (Theorem 3). If a contour passes V_1', V_1 is an $\mathcal{OV}$ and there exists a tiling line segment $\overline{V_1V_1'}$ (Theorem 1). There must have three surfaces sharing the contour segment incident with V_1. One contains $\overline{V_1V_1'}$, one contains $\overline{V_1V_2}$ and the other is on the $\mathcal{PCR}$. This violates the solid surface. So no contour passes V_1'. Same argument applies to V_2'. So no contour except $C1$ and $C2$ pass $\overline{V_1V_2'}$ or $\overline{V_1'V_2}$. Thus $\mathcal{NEC}(V_1') = \mathcal{NEC}(C2)$ and $\mathcal{NEC}(V_2') = \mathcal{NEC}(C1)$. A vertical line segment L passing the open line segment (V_1, V_2) has one intersection with the $\mathcal{PCR}$ based on Lemma 2. So the data at both ends of L must have different signs. $C1$ has different sign from the data at $(S1 \cap L)$, and $C2$ has different sign from the data at $(S2 \cap L)$. So $C1$ and $C2$ have different sign.

Suppose $C1'$ is inside $C2$. Let T be a legal tiling line segments incident with V_1 of $C1$ and V_2 of $C2$ as shown in Fig 13(b). V_1 is a negative $\mathcal{PV}$, and V_2 is a positive $\mathcal{PV}$ based on Lemma 6. With the same reasoning in the proof of Theorem 6.2, no contour except $C1$ and $C2$ pass $\overline{V_1V_2'}$ or $\overline{V_1'V_2}$. So $\mathcal{NEC}(V_2') = \mathcal{NEC}(C1)$ and $C2 = \mathcal{NEC}(C1')$. A vertical line segment L passing the open line segment (V_1, V_2) has one intersection with the $\mathcal{PCR}$ based on Lemma 2. So the data at both ends of L have different signs. $C1$ has different sign from the data at $(S1 \cap L)$, and $C2$ has the same sign from the data at $(S2 \cap L)$. So $C1$ and $C2$ have the same sign. □

proof of Theorem 7: If $C1'$ intersects $C2$, it is obvious there is a path linking $C1$ and $C2$ (Theorem 1).

Suppose condition 2 of Theorem 7 holds, Let V_1 be a negative $\mathcal{PV}$ on $C1$. We draw a line segment T, which initially is in the $\mathcal{O}(C1)$, from $V1$ to $C2$. Some contours might go across T. Because $C1$ and $C2$ are not insulated by their $\mathcal{NEC}$s. So any contour goes across T must either goes across T at an even number or intersect $C2$ as shown in Fig. 13(c). $C1$ and $C2$ have different sign. A vertical line, which passes their common outside region and is not in the inside of the sibling contours of $C1$ and C, intersects the $\mathcal{PCR}$ at one point, so it has intersection with $\mathcal{SURF}$. Thus $\mathcal{SURF}$ always has projection on T except at between $2n+1$ and $2n+2$ crossings (shown as dash line in Fig. 13(c)) where is inside the sibling contour of $C1$ or $C2$. We start from $V1$ and walk along the projection of T onto $\mathcal{SURF}$. If we encounter a contour, then we walk along that contour. It either leads to $C2$, or comes back to L. If it leads to $C2$, then we are done. If it comes back to $C2$, we again walk along the projection of L on $\mathcal{SURF}$. Because $C2$ or a contour having intersection with $C2$ is within the $2n$ to $2n+1$ crossing of L, we can always reach $C2$. So there exists a path between $C1$ to $C2$. In the case of condition 3, it is proved in the similar way. □