Robust and Efficient Surface Reconstruction from Contours

G. Cong and B. Parvin
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory
University of California
MS 50B-2239, 1 Cyclotron Road
Berkeley, CA 94720

Abstract

In this paper, we propose a new approach for surface recovery from planar sectional contours. The surface is reconstructed based on the so-called "Equal Importance Criterion", which suggests that every point in the region contributes equally to the reconstruction process. The problem is then formulated in terms of a partial differential equation, and the solution is efficiently calculated from distance transform. To make the algorithm valid for different application purposes, both the isosurface and the primitive representations of the object surface are derived. The isosurface is constructed by interpolating between sectional distance transformations. The primitive are approximated by Voronoi Diagram transformation of the surface space. Isosurfaces have the advantage that subsequent geometric analysis of the object can be easily carried out while primitives representation is easy to be visualized. The proposed technique allows for surface recovery at any desired resolution, thus, inherent problems due to correspondence, tiling, and branching are avoided.

1 Introduction

Surface reconstruction from a set of planar sectional contours has been an important problem in diverse scientific fields. These contours define the intersections of the object surface with a set of parallel planes along a desired orientation. For example, CT and MRI techniques can provide dense serial sectional representation of electron density and water molecule concentration at different locations along a particular axis. Similarly, in confocal microscopy, cross sections are obtained by focusing the optical system at specific locations along the z axis. The propose of our work is to utilize these cross sections to recover the 3D surfaces of the object for visualization as well as geometric analysis.

3D visualization methods can be classified into two categories[12]: Volume Rendering and Primitive Reconstruction methods. Volume based approaches assume that the data are available as a 3-D grid and attempt to convert directly the voxel data onto a surface. The marching cubes algorithm of Lorensen and Cline[26] or the geometrically deformed models of Miller[28] are examples of this class. The primitive based approaches, on the other hand, represent the object as an assembly of simple surface primitives, such as triangles or polygons. Once these primitives are found, they can be rendered easily from different views. Isosurfaces have the advantage that subsequent geometric analysis of the object can be easily carried out [29, 30] while primitives representation can easily exploit underlying graphic engine for rapid visualization.

Most of the existing techniques treat the "surface from contours" as a primitive reconstruction problem. The primitives are calculated from the adjacent planar contours according to their geometrical relationship. The approaches lead to three sources of ambiguities: (1) correspondence, (2) tiling, and (3) branching problems. The correspondence problem involves finding the correct connection between adjacent contours[5, 9, 10, 27, 36]. The tiling problem involves utilizing slice chords for constructing triangles in the regions between contours [20, 15, 32, 11, 17, 6, 7, 16, 31]. The main issue being how to find optimal tiling and how to account for topological changes. The branching problem occurs when a contour in one slice can correspond to more than one contour in an adjacent slice, e.g., bifurcation in the 3D shape[25, 27, 11, 6].

A few techniques aim to represent the surface as the zero-set of an implicit function[19, 24, 14] which can be visualized by the matching cubes algorithm, for example. A field function is computed in each slice and the volume data is constructed by spline interpolation of the slice images. These approaches also lead to some ambiguities: (1)field function, (2)artificial surface and (3)efficiency. How to select a appropriate field function is crucial to construct a natural smooth surface. Artificial surface arises when the zero-crossings occur at undesired locations of the implicit function. Since the zero-set can be anywhere, it becomes important to control the interpolation between slices so that no artificial surface is generated. Also, a more efficient triangulation method is needed to approximate the zero-set since matching cubes algorithm usually needs more triangles than the primitives approaches.

In this paper, we treat the problem in a entirely new way. We derive both the isosurface based and primitives based representations of the target object so that the reconstructed surface is efficient for visualization as well as geometric analysis. The approach is based on representing the problem as a partial differential equation, (PDE) which can be solved easily. The isosurface is calculated by linear interpolation between the distance transformation of adjacent contours while the primitives are computed from the Voronoi Diagram(VD). Our solution naturally avoids the correspondence, tiling, and branching problems in favor of a more robust and efficient solution. The underlying constraint is based on the Equal Importance Criterion (EIC), which suggests that all points contribute equally to the shape reconstruction process. Formally, the constraint states that surface height decreases linearly along the trajectory of its gradient, and as a result, the problem reduces to solving a PDE. Experimental results on both synthetic data and real contours are included.

1.1 Problem statement

The problem can be stated as:

Given a set of parallel planar closed contours $\mathcal{C}_i, i = 1..., m$, find the surface $\mathcal{S}$ such that $\mathcal{C}_i$ is the intersection of $\mathcal{S}$ with the plane $z = i$.

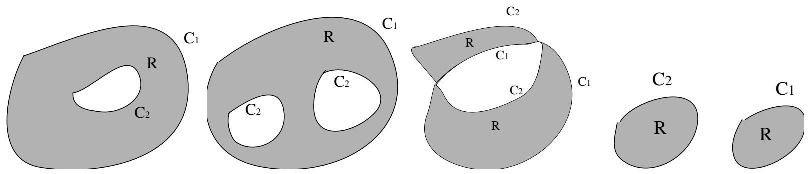
The problem can be simplified by constructing a subsurface between every two adjacent contours and defining the whole object as the union of all these subsurfaces.

Since in most cases, the contours are close to each other, smoothness of the union surface is not a problem. However, if it is necessary then a smooth operation can be easily carried out by convolving the isosurface representation with a Gaussian filter.

2 Equal Importance Criterion

The proposed reconstruction problem is underconstrained and ill-posed. To constrain the problem, we impose a smoothness measure based on EIC. Consider a pair of contours $\mathcal{C}_1$ and $\mathcal{C}_2$. Let $\mathcal{O}_i(x,y), i=1,2$ be the binary "object function" such that $\mathcal{O}_i(x,y)=1$ if the point (x,y) belongs to the object (inside), $\mathcal{O}_i(x,y)=0$ if (x,y) is on the curve, and $\mathcal{O}_i(x,y)=-1$, otherwise. The surface space $R(\mathcal{C}_1,\mathcal{C}_2)$ is defined by (see Figure 1):

$$R(\mathcal{C}_1,\mathcal{C}_2)=\{(x,y)|\mathcal{O}_1(x,y)\mathcal{O}_2(x,y)\leq 0\} \tag{1}$$



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

Figure 1: Different $R(\mathcal{C}_1,\mathcal{C}_2)$. (a)Doubly connected region. (b)Multiply connected region. (c) Intersection. (d) Isolated objects.

In $R(\mathcal{C}_1,\mathcal{C}_2)$, we want to construct a surface $f(x,y)$ such that $f(\mathcal{C}_1)=1, f(\mathcal{C}_2)=2$. Obviously, in the absence of no constraints, infinitely many solutions exist. To constrain the problem, we assert that *every point in $R(\mathcal{C}_1,\mathcal{C}_2)$ is equally important and contributes similarly to the reconstruction process. Any other assumption means that we know something about the surface.* We call this the *Equal Importance Criterion.* This constraint is formalized by requiring that the change in the gradient-magnitude along the gradient direction should be zero, that is:

$$\mathcal{J}_1(f) = \nabla(|\nabla f|) \cdot \frac{\nabla f}{|\nabla f|} = 0 \tag{2}$$

where ∇f indicates the gradient of f, $|\,|$ the norm and $\cdot$ the inner product. The above PDE implies that along each trajectory of the gradient of the surface, the magnitude of the gradient is a constant. In another words, the height decreases linearly from 2 to 1. The level curves of the surface are equally distributed along the gradient direction. Thus, in view of height, which is our only evidence about surface, all points are equally important to us.

$\mathcal{J}_1$ can be reduced to:

$$\mathcal{J}_2(f) = f_x^2 f_{xx} + 2 f_x f_y f_{xy} + f_y^2 f_{yy} = 0 \tag{3}$$

where subscript indicates derivative, such as $f_x = \frac{\partial f}{\partial x}$, $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$. Thus,

$$\begin{aligned} \mathcal{J}_2(f) = 0, &\quad (x, y) \in R(\mathcal{C}_1, \mathcal{C}_2) \\ s.t.\ \ f(\mathcal{C}_1) = 1, &\quad f(\mathcal{C}_2) = 2. \end{aligned} \tag{4}$$

Equation (3) is called the "Infinity Laplacian" (see Appendix A), which has been studied in the literature [1, 2, 3, 13]. They showed that Equation (4) has at most one solution which may not be smooth for any $\mathcal{C}_1$ and $\mathcal{C}_2$. But if we redefine the "solution" in a suitable weak sense, then a solution continuous at least to C^1 does exist.

3 Isosurface reconstruction

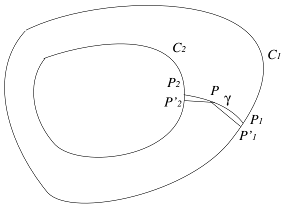
In our PDE based approach, the correspondence, tiling, and branching problems have been eliminated and the distance between $\mathcal{C}_1$ and $\mathcal{C}_2$ in the z direction is no longer important because it only changes the solution by a scale. We now develop an efficient solution for the above equation.

3.1 Solving the PDE

Many numerical methods can be used to solve Equation (4), and at least a weak solution is guaranteed. However, a more straightforward and efficient approximation is possible.

Let's define $\mathcal{D}_i(x, y), i = 1, 2$ as the *Distance Transformation* of curve $\mathcal{C}_i$, where $\mathcal{D}_i(x, y)$ has the same sign of $\mathcal{O}_i(x, y)$. For each point p (shown in Figure 2), there should be a gradient trajectory γ passing through it such that it intersects $\mathcal{C}_1$ and $\mathcal{C}_2$ at p_1 and p_2, respectively. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal height contours, it is easy to show that the normal of these two contours and the gradient of surface are in the same direction. Thus $\gamma \perp \mathcal{C}_1$ at p_1 and $\gamma \perp \mathcal{C}_2$ at p_2. We can approximate the curve γ, passing through p, by drawing two line segments $pp'_1 \perp \mathcal{C}_1, pp'_2 \perp \mathcal{C}_2$, to create $p'_1pp'_2$. Let l denote the length of γ from p_1 to p_2. Hence, $l \approx |p'_1p| + |p'_2p|$. The preceding formulation indicates that $|p'_1p| = -\mathcal{D}_1(p)$, $|p'_2p| = \mathcal{D}_2(p)$. Since the height decreases linearly, f can be approximated by:

$$f(x, y) = \frac{2|p'_1p| + |p'_2p|}{|p'_1p| + |p'_2p|} = \frac{\mathcal{D}_2(x, y) - 2\mathcal{D}_1(x, y)}{\mathcal{D}_2(x, y) - \mathcal{D}_1(x, y)} \tag{5}$$



(a)

Figure 2: Approximation solution of the PDE.

3.2 Isosurface construction

The 3D isosurface representation, $\phi(x, y, z), 1 \leq z \leq 2$, can now be expressed as:

$$\phi(x, y, z) = (z - 1)\mathcal{D}_2(x, y) + (2 - z)\mathcal{D}_1(x, y) \tag{6}$$

This (zero value) isosurface $\phi(x, y, z) = 0$ is:

$$z(x, y) = \frac{\mathcal{D}_2(x, y) - 2\mathcal{D}_1(x, y)}{\mathcal{D}_2(x, y) - \mathcal{D}_1(x, y)} \tag{7}$$

which is exactly the surface that we reconstructed in (5). From equation (7), since $1 \leq z \leq 2$ if and only if $\mathcal{D}_2(x,y)\mathcal{D}_1(x,y) \leq 0$, $\phi(x,y,z) = 0$ occurs only in the region $R(\mathcal{C}_1, \mathcal{C}_2)$. Thus, we cannot get an artificial isosurface. Figure 3 show the reconstruction results for different $\mathcal{C}_1, \mathcal{C}_2$ combinations.

The proposed method can be applied iteratively to every pair of adjacent contours $\mathcal{C}_i, \mathcal{C}_{i+1}, i = 1, ..., m-1$ for constructing a series of subsurfaces $\mathcal{S}_i$. These subsurfaces, $\mathcal{S}_i : i = 1, ..., m-1$, form the whole surface, namely $\mathcal{S} = \bigcup_i \mathcal{S}_i$. The final output of the algorithm is a 3D data with new slices inserted between every $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$. Suppose that we want a resolution of $\sigma < 1$, say 0.1, along the z direction, then reconstruction should include between each pair of adjacent contours $Z-1$ new slices with $Z = \frac{1}{\sigma}$(an integer Z is expected).

3.3 Smoothing

In most cases, the contours are close to each other, thus, the smoothness of the union surface is not a problem. When the contours are considerably apart from each other, the surface may be not smooth at the contour locations. The simplest way to smooth the surface is to convolve $\phi(x,y,z)$ with a small scale 3D Gaussian filter, which is well known as equivalent to move every point on the surface along its normal direction at a speed of its mean curvature[21]. Those points or regions with high curvatures will be smoothed.

Suppose that the smoothed volume is represented by $\phi' = \phi * G$, where G is the Gaussian filter, the derivatives along the smoothed isosurface $\phi'(x,y,z) = 0$ can be calculated from the derivatives of ϕ'. For example,

$$
\begin{aligned}
z_x &= -\frac{\phi'_x}{\phi'_z} \\
z_y &= -\frac{\phi'_y}{\phi'_z} \\
z_{xx} &= -\frac{\phi'_{xx}{\phi'_z}^2 - 2\phi'_{xz}\phi'_x\phi'_z + \phi'_{zz}{\phi'_x}^2}{{\phi'_z}^3} \\
z_{yy} &= -\frac{\phi'_{yy}{\phi'_z}^2 - 2\phi'_{yz}\phi'_y\phi'_z + \phi'_{zz}{\phi'_y}^2}{{\phi'_z}^3}
\end{aligned}
$$

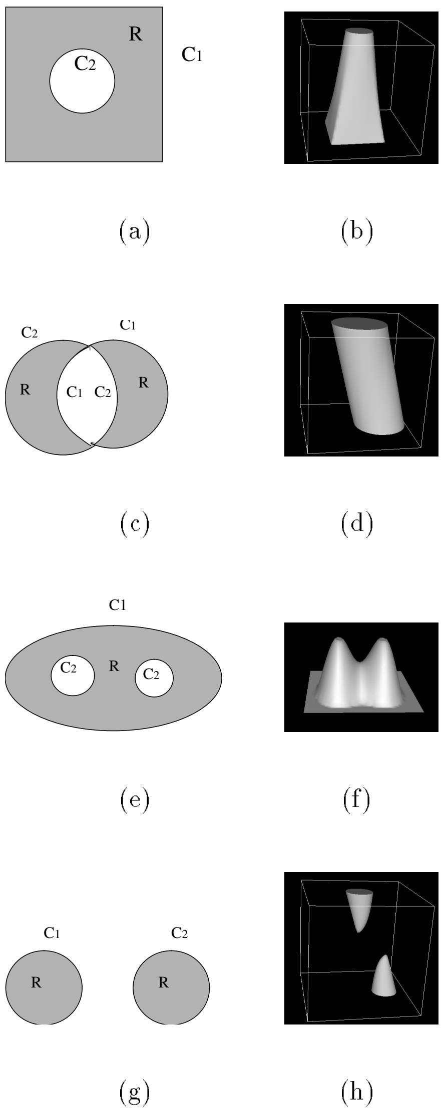


Figure 3: Isosurface reconstructed from two contours. (a)(c)(e)(g) Contours; (b)(d)(f)(h) Computed isosurface.

$$z_{yy} = -\frac{\phi'_{xy}{\phi'_z}^2 - \phi'_{xz}\phi'_y\phi'_z - \phi'_{yz}\phi'_x\phi'_z + \phi'_{zz}\phi'_x\phi'_y}{{\phi'_z}^3}$$

And the derivatives of the smoothed isosurface can be computed as:

$$\begin{aligned}
\phi'_x &= \phi * G_x \\
\phi'_y &= \phi * G_y \\
\phi'_{xx} &= \phi * G_{xx} \\
\phi'_{xy} &= \phi * G_{xy} \\
\phi'_{yy} &= \phi * G_{yy}
\end{aligned}$$

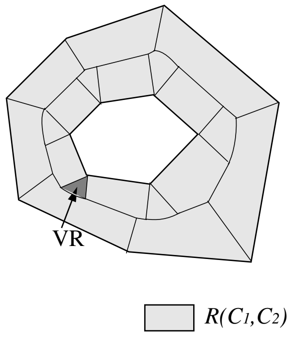
Since Gaussian is infinitely differentiable, the corresponding smoothed isosurface is infinitely continuous.

4 Primitive representation

A 3D triangle is the basic surface patch used in most visualization systems. In this section, we show how to approximate the surface define by (5) as a assembly of triangles. The basic idea is to partition $R(\mathcal{C}_1, \mathcal{C}_2)$ into 2D triangles and then project them to 3D space.

4.1 Partitioning $R(\mathcal{C}_1, \mathcal{C}_2)$

Partitioning is based on the Voronoi Diagram (VD), one of the most fundamental data structures in computational geometry and computer vision[4, 23, 22, 34]. Like most of previous works[5, 10, 27, 36], we assume that $\mathcal{C}_1, \mathcal{C}_2$ are polygonal curves. The vertices and the segments linking them are called elements. The VD is a set of points inside $R(\mathcal{C}_1, \mathcal{C}_2)$, where each of them has at least two closest elements equidistant to it. $R(\mathcal{C}_1, \mathcal{C}_2)$ is divided by its VD into singly-connected regions, called Voronoi Regions (VR), according to the nearest-neighbor-rule. Each point in $R(\mathcal{C}_1, \mathcal{C}_2)$ is associated with the element closest to it, and all the points in one VR have the same closest element. See Figure 4. The "net-like" VD consists of line segments and parabola that are fitted by polygon curves [18, 33, 35]. Thus each VR becomes a polygon.



(a)

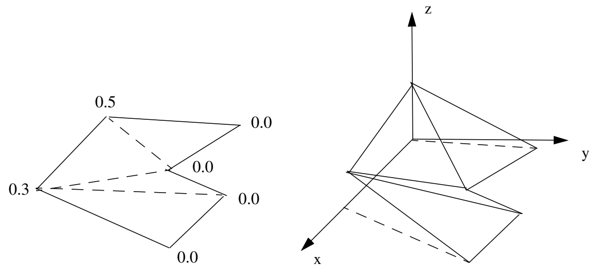
Figure 4: Voronoi diagram.

An iterative approach then partitions each VR into triangles. The approach begins by randomly finding two non-sequential vertices, in one VR, that when linked by a segment, the segment is totally inside the VR. This segment divides the VR into two parts. For each part, if it is a triangle, the algorithm returns. Otherwise, this part is set as a new polygon and the process continues recursively.

4.2 Reconstructing 3D triangles

The computed 2D triangles are then projected into 3D space, and the corresponding z-values of the vertices (of the triangles) are calculated by (5). A example is shown in Figure 5.

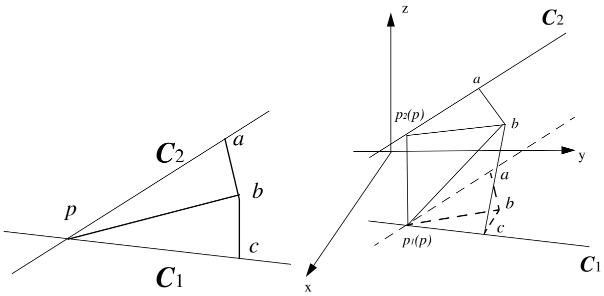
Two specific situations need more careful treatment. First, if $\mathcal{C}_1$ and $\mathcal{C}_2$ intersect at point p then the z-value of p can be either 1 or 2, as shown in Figure 6. In 2D, let $p = (x, y)$. In 3D, p is denoted by $p_1 = (x, y, 1)$ on $\mathcal{C}_1$ and $p_2 = (x, y, 2)$ on $\mathcal{C}_2$. In this case, an additional triangle $b - p_1 - p_2$ besides triangles $a - b - p_2$ and $b - c - p_1$ must be constructed to preserve the continuity of surface. Second, if $\mathcal{C}_1$ and $\mathcal{C}_2$ share a common segment, as shown in Figure 7, the z-value of that segment can also be either 1 or 2. In 2D, let $a = (x_1, y_1)$, $b = (x_2, y_2)$. In 3D, a and b are denoted by $a_1 = (x_1, y_1, 1), a_2 = (x_1, y_1, 2), b_1 = (x_2, y_2, 1), b_2 = (x_2, y_2, 2)$, respectively. Hence, a



(a) (b)

Figure 5: 3D triangle reconstruction. (a) 2D triangles. (b) 3D patches.

rectangle $a_1 - a_2 - b_2 - b_1$ must be constructed.



(a) (b)

Figure 6: $\mathcal{C}_1, \mathcal{C}_2$ intersect at point p. (a)2D partition; (b)3D triangles.

Figure 8 shows the reconstruction results for different $\mathcal{C}_1$ and $\mathcal{C}_2$ combinations.

5 Experimental results

The proposed protocol has been tested on real medical images for both the isosurface representation(Figure 9 and Figure 10) and the primitives representation(Figure 11).

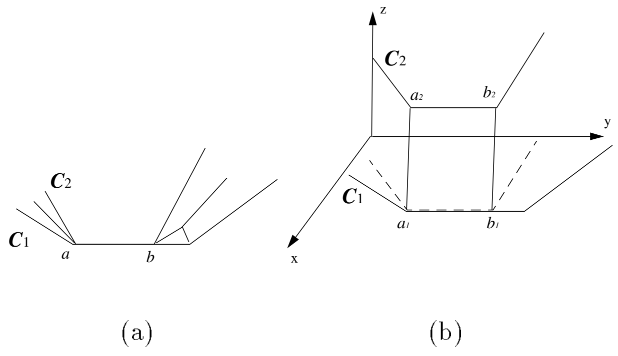


Figure 7: $\mathcal{C}_1, \mathcal{C}_2$ share a common segment. (a)2D partition; (b) 3D rectangle.

In Figure 9, the reconstruction results are computed from segmented magnetic resonance images of a patient with edema. This figure shows the white matter corresponding to the cerebral cortex(light region), as well as damaged area due to edema(dark region). The original data is 256 by 256 and includes 22 slices. Three new slices have been reconstructed from each pair of adjacent slices. No post-smoothing is performed on these data. (g)(h) are close views of the surface. As we can see, our approach treats topological changes naturally and smoothly.

In Figure 10, we show the smoothing effect of Gaussian filter. (a)(c) are the original isosurfaces correspondent to the region of edema in Figure 9; (b)(d) are the isosurfaces smoothed by Gaussian filter with scale 1. The advantage of such a smooth process are that we do not need a parameter representation of the surface and the results are adaptive for that small features are smoothed out while larger features are preserved. The distance transformations in the above experiments are computed by the method proposed in [8].

In Figure 11, we present the results of VD based triangle approximation of the surface. The original CT data is 64×64 with 64 slices. The polygon contours are obtained by thresholding each of the slices followed by polygonal approximation[18, 33] with certain resolution. We then use only part of the contours to reconstruct the surface. The triangles constructed as in section 4 are visualized directly. For each pair of contours, suppose that there are n vertices among which m vertices are concave,

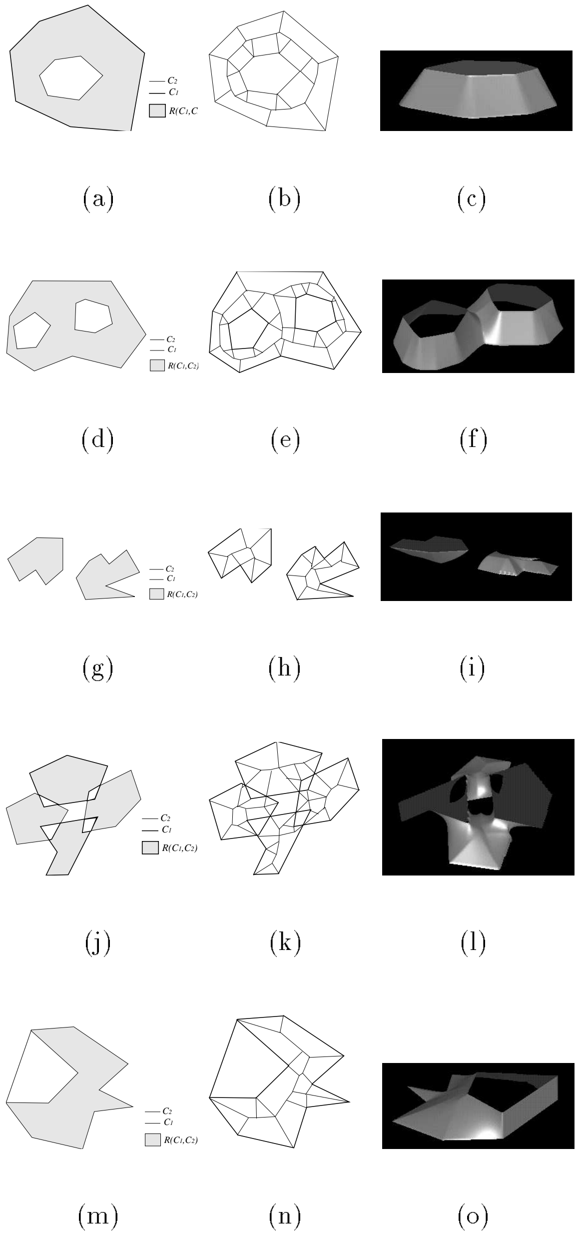


Figure 8: Surface reconstructed from two contours. (a)(d)(g)(j)(m) Contours; (b)(e)(h)(k)(n) Approximated VD; (c)(f)(i)(l)(o) Computed surface.

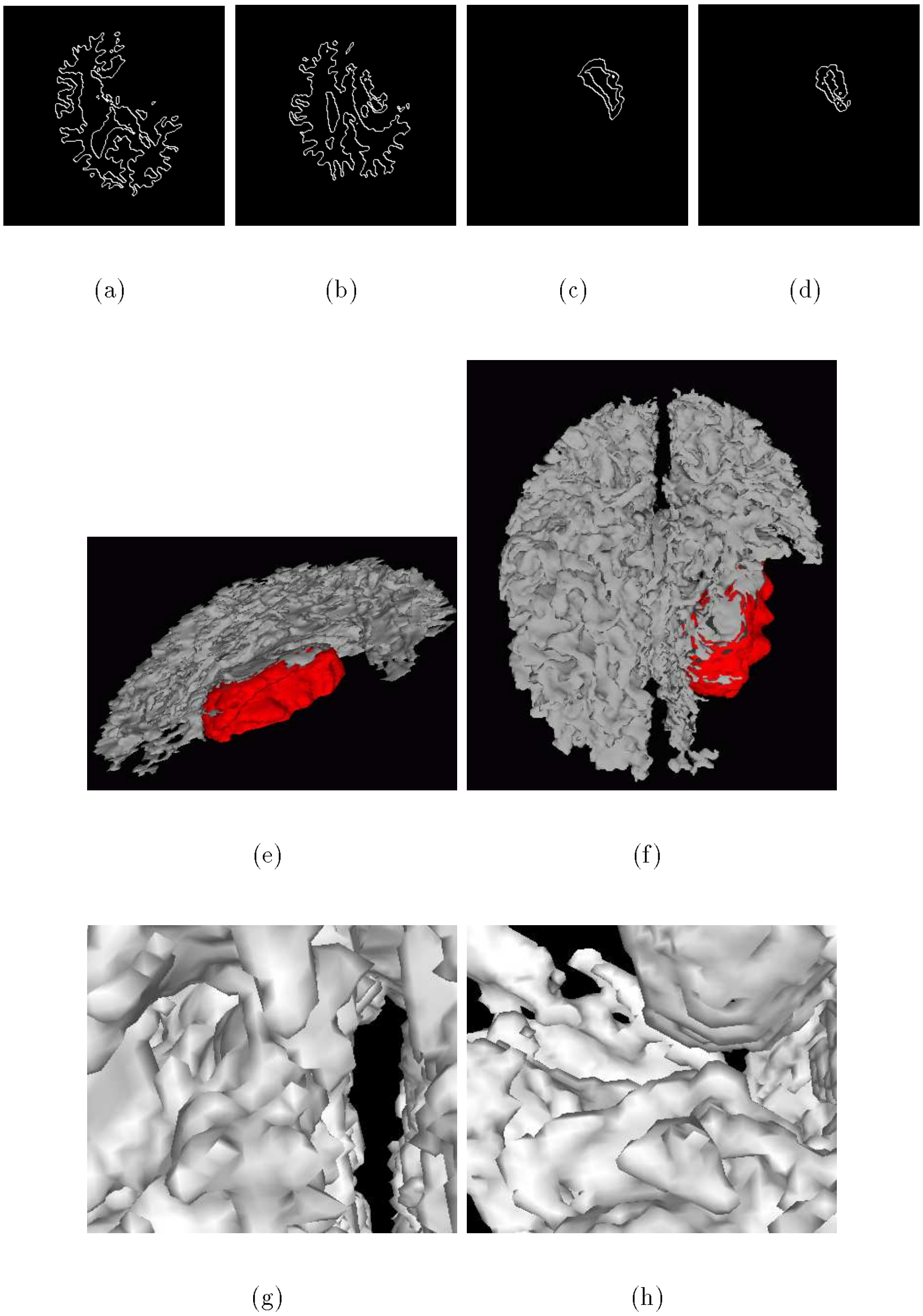
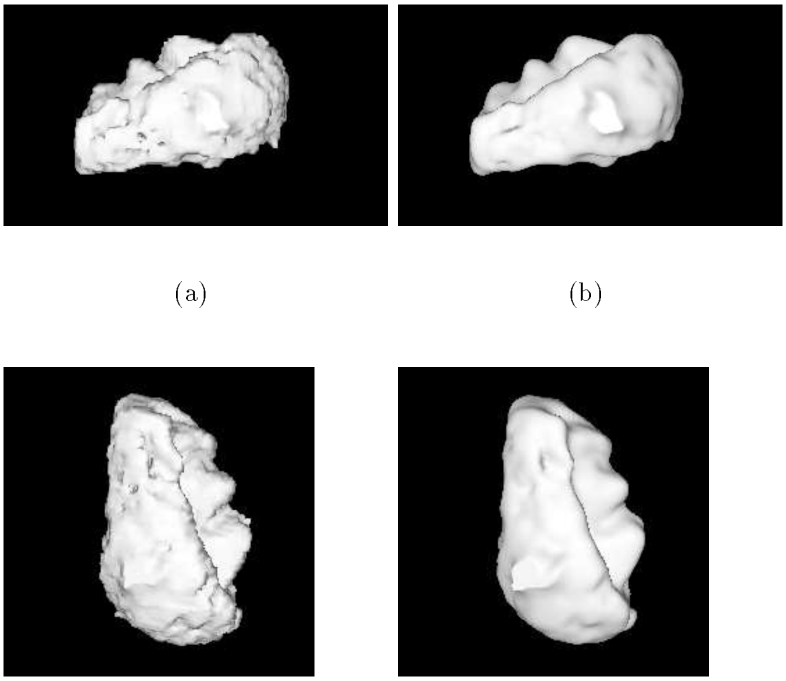


Figure 9: Reconstruction results of white matter in cortex and region due to edema: (a)(b)(c)(d) Contours from segmented cortex and edema; (e)(f)Two views of surfaces. (g)(h)Tow close views of the white matter.



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

Figure 10: Smoothing results of the region due to edema: (a)(c)Original isosurface; (b)(d)Smoothed by Gaussian filter with scale 1;

$R(\mathcal{C}_1, \mathcal{C}_2)$ is then partitioned into $m+n$ VRs[22]. Each VR are further divided into several triangles. In our experiments, the average number of triangles in one VR is 2.7. Thus, the total number of triangles for one pair of contours is approximately $2.7(m+n)$ and the computation complexity is $O(m+n)$. The VD are computed based on the method proposed in [22].

6 Conclusion

Shape from cross sectional contours is an important problem in diverse fields of science, and has been studied extensively. However, most (if not all) of these methods suffer from correspondence, tiling, and branching problems. Furthermore, previous solutions have been limited to variants of triangulation that are scale dependent, with unpredictable behavior near the saddle points. The novelty of the proposed method is in its unique smoothness measure, the corresponding PDE, and its simple solution based on distance transform. We showed that a linear solution provides an adequate representation of isosurface. In the case of primitives representation, VD gives us a natural segmentation of the surface space and enables us to construct small surface patches more easily for any shape. We have tested and verified our approach on data with different degrees of complexities, ranging from simple geometric features to complex and convoluted structure of cortex.

A A note on infinity Laplacian

This equation is the limit for $p \to \infty$ of the so-called p-Laplace equation, whose solution f minimize the L_p norm of the gradient among functions ϕ satisfying the given boundary values of $R(\mathcal{C}_1, \mathcal{C}_2)$:

$$L_p(\phi) = \left(\int_{R(\mathcal{C}_1,\mathcal{C}_2)} | \bigtriangledown \phi|^{2p} dxdy\right)^{\frac{1}{2p}}, \quad p \to \infty \tag{8}$$

In the limit the L_p norm turns into the supermum norm. Formally the solutions of this equation are the "minimizers of the sup norm of the gradient"[1].

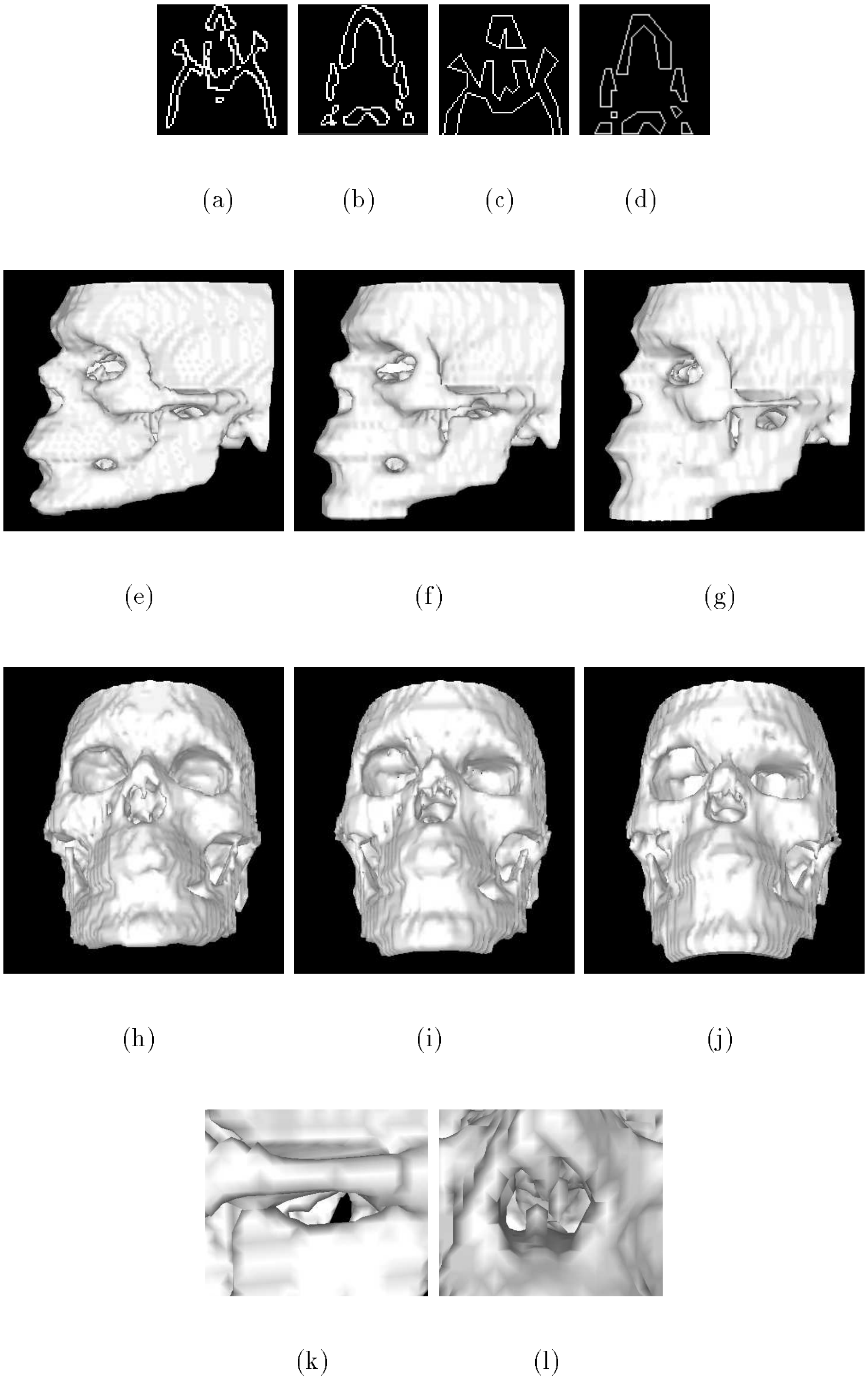


Figure 11: Surface reconstructed from CT data. (a)(b)High resolution contour number 24 and 42; (c)(d)Low resolution contour number 24 and 42; (e)(f) Surface reconstructed from 22 high resolution contours; (g)(h) Surface reconstructed from 16 high resolution contours; (i)(j) Surface reconstructed from 22 low resolution contours; (k)(l) Two close views of (e).

References

[1] G. Aronsson. Extension of function satisfying lipschitz conditions. *Arikv för Matematik*, 6:551–561, 1967.

[2] G. Aronsson. On the partial differential equation $u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy} = 0$. *Arikv för Matematik*, 7:133–151, 1968.

[3] G. Aronsson. On certain singular solutions of the partial differential equation $u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy} = 0$. *Manuscripta Math*, 41:133–151, 1981.

[4] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23:343 – 405, 1991.

[5] C. Bajaj, E. Coyle, and K. Lin. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, 58(6):524–543, 1996.

[6] J. Boissannat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 44(1):1–29, 1988.

[7] J. Boissannat and B. Geiger. Thre dimensionsl reconstruction of complex shape based on the delaunay triangulation. *INRIA, Tech. Rep.*, 1992.

[8] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[9] Y. Bresler, J. Fessler, and Macovski. A bayesian approach to reconstruction from imcomplete projections of a multiple object 3d domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):840–858, 1988.

[10] R. Durikovic, T. Yauchi, K. Kaneda, and H. Yamashita. Shape-based calculation and visualization of general cross-section through biological data. In *Proceedings of International Conference on Information Visualization*, 1997.

[11] A. Ekoule, F. Peyrin, and C. Odet. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM transactions on graphics*, 10(2):182–199, 1991.

[12] T. Elvins. A survey of algorithm for volume visualization. *Computer Graphics*, 26:194–201, 1992.

[13] L. Evans. Estimates for smooth absolutely minimizing lipschitz extensions. *Electronic*
Journal of Differential Equations, http://ejde.math.swt.edu/Volumes/1993/03-Evans/abstr.html, 1993(3):1–9, 1993.

[14] M. Floater and Westgaard. Smooth surface reconstruction from cross-sections using implicit methods. *SINTEF Rep.*, 1996.

[15] H Fuchs, Z. Kedem, and S. Uselton. Optimal surface reconstruction from planar contours. *Communications of the ACM*, 20(10):693–702, 1977.

[16] B. Geiger. Three dimensional modeling of human organs and its application to diagnosis and surgical planning. *INRIA, Tech. Rep.*, 1993.

[17] G. Gitlin, J. O'Rourke, and V. De Pigueiredo. On reconstruting polyhedra from parallel slices. *International Journal of Computational Geometry and Application*, 6(1):103–122, 1996.

[18] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.

[19] M. Jones and M. Chen. A new approach to the construction of surfaces from contours data. *Computer Graphics Forum*, 13:75–84, 1994.

[20] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM Journal of Research and Development*, 19:2–11, 1975.

[21] B. Kimia, Tannenbaum, and S. Zucker. On the evolution of curves via a function of curvature, i: The classic case. *JMAA*, 163(2), 1992.

[22] D. Lee. Media axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:363–369, 1981.

[23] D. Lee and R. Drysdale. Generation of voronoi diagrams in the plane. *SIAM J. Comput.*, 10:73–87, 1981.

[24] D. Levin. Multidimensional reconstruction by set-valued approximation. *IMA J. Numberical Analysis*, 6:173–184, 1986.

[25] W. Lin and S. Chen. A new surface interpolation technuque for reconstructing 3d objects from serial cross-sections. *Computer Vision, Graphics, and Image Processing*, 48:124–143, 1989.

[26] W. Lorensen and H. Cline. Match cubes: a high resolution 3d surface reconstruction algorithm. *Computer Graphics*, 21:163–169, 1987.

[27] D. Meyers, S. Skinner, and K. Sloan. Surface from contours. *ACM transactions on graphics*, 11(3):228–258, 1992.

[28] J. Miller, D. Brean, W. Lorensen, R. O'Bara, and M. Wozny. Geomotrically deformed models: a method for extracting closed geometric models from volume data. *Computer Graphics*, 25:217–226, 1991.

[29] O. Monga, N. Ayache, and P. Sander. From voxel to intrinsic surface features. *Image and Vision Computing*, 10(6), 1992.

[30] O. Monga, S. Benayoun, and O. Faugeras. From partial derivatives of 3d density images to ridge lines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 354–359, 1992.

[31] J. Oliva, M. Perrin, and S. Coquillart. 3d reconstruction of complex polyhedral shapes from contours using simplified generalized voronoi diagram. *Proceedings of Eurographics*, 15:397–408, 1996.

[32] Y. SHinagawa and T. Kunii. The homotopy model: A generalized model for smooth surface generation from cross sectional data. *The Visual Computer*, 7(2-3):72–86, 1991.

[33] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing analysis and Machine Vision.* Chapman & Hall, 1995.

[34] V. SriniVasan and L. Nackman. Voronoi diagram for multiply-connected polygonal domains i : Algorithm. *IBM Journal of Research and Development*, 31:361–381, 1987.

[35] W. Wang. Banary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, 31(10):1502–1524, 1998.

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