A Simple and Optimal Energy Surface Reconstruction Algorithm from Volumetric Data

Kexue Liu
Ramani Duraiswami
Larry Davis
University of Maryland Institute for Advanced Computer Study
College Park, MD 20742

February 4, 2003

Abstract

We describe a new method for building 3D surface meshes from volumetric images, as are created in Magnetic Resonance Imaging (MRI). The method works by first extracting the contours from the MRI data, and then using modified polygon boolean operations to find difference polygons. The last step is performed by applying a Constrained Delauny Triangulation (CDT) algorithm to build a surface mesh. Since it is based on CDT, the mesh quality is usually very good. Moreover, the resulting triangular surface mesh is the optimal one for surface interpolation. In case of undersampled images, the mesh quality can be improved dramatically by inserting new Steiner points in to the mesh. Thus the mesh can be made suitable for use in scientific computation. Our method has the advantages of combining the generation and optimization of the mesh in one step compared with other methods. These advantages are illustrated.

1 Introduction

In applications such as medical imaging, surgery planning, thin-section tissue imaging, reverse engineering, or machine diagnostics, it is often necessary to recreate three dimensional surface models from volumetric data sets that consist of a series of parallel planar cross-sections. The cross-sections (slices), are obtained by Computerized Tomography (CT) [1], Magnetic Resonance Imaging (MRI) [2], or in some cases the structure is itself physically sliced (transmission electron microscopy of tissue) [3]. Three main approaches have been developed to solve this problem: Volume reconstruction, Surface reconstruction, and Deformable Models.

The first of these techniques is inspired by image processing techniques and treats the data as 3D images obtained by piling up the images of cross-sections. The interpolation is done in the vicinity of each voxel, or is based on a pillar of voxels of all slices with the same (x, y) coordinates. When the available data form a dense 3-D lattice of values, a volume based approach such the as marching cubes algorithm of Lorensen and Cline [17] as suggested by Meyers et al [7], is often used. The main disadvantage of this approach is its large storage requirement. Deformable Model, is created by placing a 'seed' model in the volumetric data set. The model is then deformed by a process that minimizes a cost function [6, 11]. This approach provides users with great flexibility to effectively interact with the dataset. The continuity and smoothness of the model can compensate for the unwanted sampling artifacts and irregularities introduced at the object boundaries [11]. The problem related to deformable model method is how to choose weighing parameters of the cost function: there is no unified way to determine the parameters.

Surface reconstruction, which constructs a polyhedral model of the object that interpolates the boundary of the cross-sections. More precisely, given a series of parallel planar contours, each consisting of a collection of non-crossing, but possibly nested, closed and simple polygonal curves, construct a polyhedral solid model whose cross sections along the given planes coincide with the input contours. The reconstruction process is preceded by an edge-detection process. There are vast numbers of papers in the literature using this approach to reconstruct the surface. The following is a brief review of this approach.

2 Previous Work

The problem of generating a surface from a set of planar contours has three sub-problems (see Meyers [7]) :
The *Correspondence Problem* requires matching like portions of the contours and is solved by determining the topological adjacency relationships between the contours of a data set. A solution to the correspondence problem determines the coarse topology of the final surface.
The *tiling problem* is to determine the point correspondence and solved by generating the “best” topological adjacency relationships between the points on pairs of contours from adjacent sections by constructing a triangular mesh from their points.
The *branching problem* arises when an object is represented by a different number of contours in adjacent sections, in which case, the standard method for solving the tiling problem can not be used directly. A solution to the tiling problem produces a detailed description of the geometry of the reconstructed surface.
The solution to a surface reconstruction problem is not uniquely defined, especially when the distances between cross sections are large. In this case, there is no general method to solve the correspondence problem. In Figure 1, the $x - y$ projections are far away from each other. There are at least two distinct ways to triangulate these two polygons. One way is to regard them as the last section of one component and the first section of another component. Another way is to treat them as two sections of one component. In our approach we usually regard them as the first case. If the results are not satisfying, our approach fits consecutive contours into an implicit function as described in Turk et al [20], and then introduces intermediate sections to make sure that the new consecutive sections are close enough.

Another problem in surface reconstruction is that it is not always possible to find an interpolating polyhedron between two polygons that lie in two parallel planes. That is, any attempted interpolation produces a self-intersecting surface. This holds only when the interpolating triangles are all assumed to have two vertices from one polygon and the third from another polygon. Gitlin et al [8] give such an example in their work. Our approach does allow the triangles whose vertices are in the same planes, which means all three vertices of some of the triangles belong to the same polygon.

Fuchs et al [5] may have been the first to reconstruct optimal surfaces from planar contours. They reduce the surface reconstruction problem to

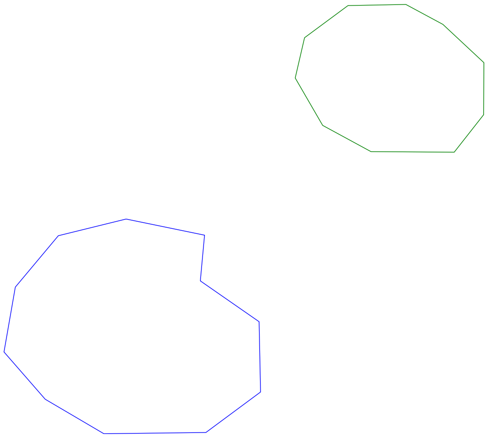

Figure 1: Contour Branching

the problem of finding certain minimum cost cycles in a directed graph. Boissonnat [12] used Delaunay triangulation for each slice, projected one triangulation onto the other, and obtained a collection of tetrahedra, aiming to maximize the sum of their volumes. The approach in this paper is close related to that of Barquet [16]. This algorithm uses a partial curve matching technique for matching parts of the contours, an optimal triangulation of 3D polygons for resolving the unmatched parts, and a minimum spanning tree heuristic for interpolating between non simply connected regions. Cong et al [21] project consecutive contours to the $x - y$ plane, in the domain of all difference polygons, formulate the problem in terms of partial differential equations (Infinity Laplacian), and a simple linear solution is calculated from the distance transform. This solution is the “minimizer of the sup norm of the gradient of the function”. In the sense of minimization, our approach is very similiar to this. Since our approach provides a “ minimizer of all the piecewise linear interpolations induced by triangulation”.

3 Slices of volumetric data and contours

We extract the edges of each slice through an edge detector [26]. For each inner boundary contour, the points are sorted in clockwise order, for each outer bounday contour, the points are sorted in anticlockwise order. In such a way, we can easily differentiate the inner boundary contours and the outer

Figure 2: MRI images and the corresponding contours

boundary contours. Thus it provides us a way to find the difference polygons. Figure 2 shows some contours obtained from an edge detection process.

4 From contours to meshes

The contours obtained from the MRI images captures almost all of the details of the pinna mold, which is a man-made human outer ear model. (See Figure 3). Before we describe the algorithm, for completeness, let us briefly review some basic graphics concepts.

4.1 Delaunay Triangulation, Constraint Delaunay Triangulaton, Voronoi Digram

A **Delaunay triangulation** of a point set is a triangulation of a point set with the property that no point in the set falls in the interior of the circumcircle of any triangle in the triangulation. It has the property that the minimal angle of all the triangles is maximized.
A **Voronoi diagram** of a point set in the plane is a subdivision of the plane into polygonal regions (some of which may be infinite), where each region is the set of points in the plane that are closer to some input point than to any other input point. Suppose that $S = \{p_i \in \mathbb{R}^2 | i = 1, \ldots, N\}$ is a point set, $V(i) = \{q \in \mathbb{R}^2 | dist(q, p_i) \leq dist(q, p_j), \forall 1 \leq j \leq N\}$ is the set of points closer to site p_i than to any other site in S, $V(i)$ is called the **Voronoi cell** associated with p_i. The union over all $V(i)$, $V(S)$ is the **Voronoi Diagram**.

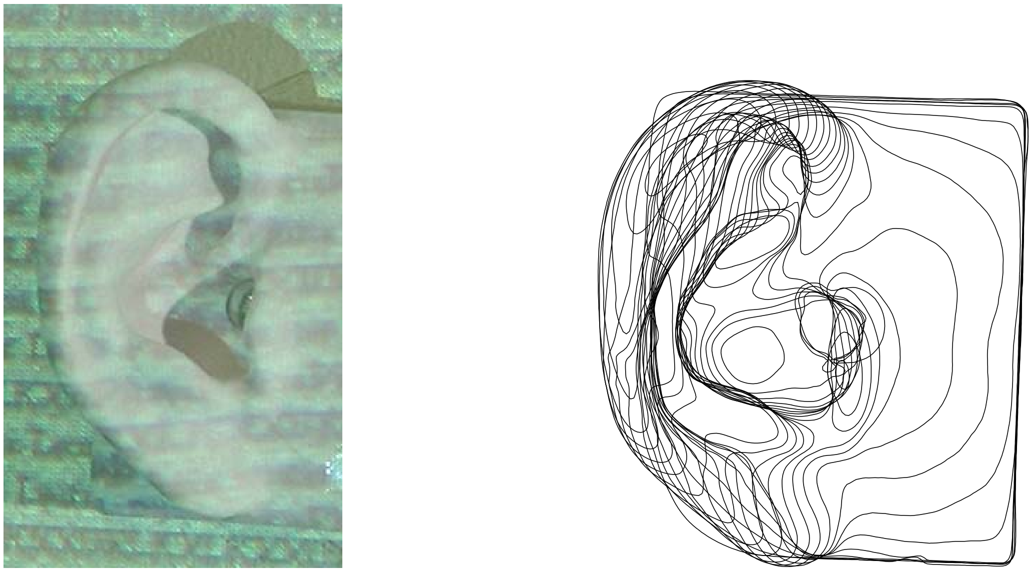

Figure 3: pinna mold image and contours

The Voronoi diagram is the geometric dual of the Delaunay triangulation.
A **Planar Straight Line Graph (PSLG)** is a collection of points and segments. Segments are edges whose endpoints are points in the PSLG, and whose presence in any mesh generated from the PSLG is enforced.
A **Constrained Delaunay Triangulation (CDT)** of a PSLG is similar to a Delaunay triangulation, but each PSLG segment is present as a single edge in the triangulation. A CDT is not a Delaunay triangulation.
Steiner Points refer to points which are added to the set of vertices of the input PSLG.

4.2 Modified Polygon Boolean Operations

Suppose that we have polygons A and B, the new polygons which are constructed with points either in A or in B, but not both, are called **difference polygons** [13]. The difference polygons of polygon $A_1, A_2, ..., A_{11}$ and polygon $B_1, B_2, ..., B_{10}$ are $C_1, A_1, ..., A_8, B_5, B_4, ..., B_2$ and $B_1, C_1, A_{11}, A_{10}, A_9, B_6, ..., B_{10}$. But the modified difference polygons are formed by skipping the intersecting edges. The modified difference polygons are the shaded polygons in Figure 4. The intersecting edges will be treated differently in the mesh building process.

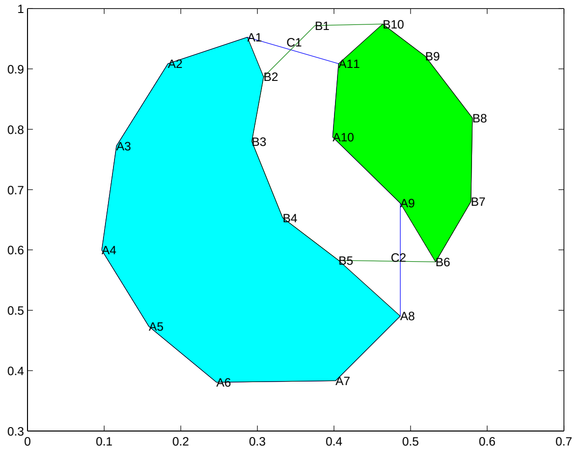

Figure 4: Polygon boolean operations, the shaded area are the modified difference polygons.

4.3 Constrained Delaunay Triangulation based approach

Algorithm

1. Project the contours in adjacent cross-sections onto the same $x - y$ plane. In practice, we simply ignore the z coordinates at the moment.
2. Find the corresponding contours in the adjacent cross-section for every contour in the current cross-section. The contours matches could be one-to- one, one-to-many, or many-to-one. We use a distance rule to match contours. If the distance between two contours from different cross-sections which have the same orientation is less than some threshold, we regard them as corresponding contours. In this step, the orientation of the contours is very important. It is not necessary to match contours which have different orientations.
3. For groups of corresponding contours, use modified polygon boolean operations to find the difference polygons. The resulting polygons could be polygons with holes or simple polygons. For each resulting polygon, use constrained Delaunay triangulation to triangulate it. For the remaining intersecting edges, we build the mesh by optimizing the angles

of all triangles. Usually, there are only two possibilities, by comparing the inner angles of the triangulation, we decide which choice is better.

4. Project 2D points back to 3D points by adding z coordinates back to all of the points, the connections remain the same. Because of the nice property of Constrained Delaunay Triangulation, the triangles of the resulting 3D mesh are very good.

4.4 Undersampled Data

If two polygons in two consecutive sections have the same orientation, but their $x-y$ plane projections have nothing in common, we usually don't treat them as the same component of the object. But if they do, the best way to deal with this situation is to fit them to a variational implicit function as Turk et al [20]and Beatson et al [25] do. We then introduce intermediate sections between the original sections and make sure that the $x-y$ projections of two adjacent sections intersect with each other. The basic idea of variational interpolation is using the appropriate radial basis function to express the interpolation function as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \phi(|\mathbf{x} - \mathbf{x}_i|) + P(\mathbf{x}) \tag{1}$$

Where the ϕs are chosen th satisfy some minimum energy criterion (e.g. thin plate splines). Here λ_i are coefficients and $P(\mathbf{x})$ is a global polynomial function. Based on the point set and their normals, one can derive a linear system. The solution of it gives λ_i as well as the coefficients of the polynomial $P(\mathbf{x})$. It is worth mentioning that this solution has the minimum energy:

$$\int \int_{\Omega} f_{xx}^2(\mathbf{x}) + 2f_{xy}^2(\mathbf{x}) + f_{yy}^2(\mathbf{x}) \tag{2}$$

if we use the biharmonic radial basis function $\phi(|\mathbf{x}|) = |\mathbf{x}|$.

4.5 Mesh Refinement

Due to limitations in volumetric data sampling, many other surface reconstruction algorithms fail to produce quality triangle meshes. They need some post processings (such as remeshing, or optimizing) techniques to improve the quality of the mesh. Without introducing steiner points, sometimes, the quality of the mesh generated by Constrained Delaunay Triangulation is not good either. Because of the inherent characteritics of CDT (we use an incremental constrained Delaunay triangulation algorithm), we can easily introduce steiner points into the mesh as Ruppert [15] and Shewchuk [14] do. In our implementation, if the ratios of the edge lengths of one triangle are over some threshold, we insert the circumcenter of this triangle (3D) into the vertices and retriangulate the projection of new points set using (2D) Constrainted Delaunay Triangulation. It is easy to show that the projection of the circumcenter of a 3D triangle is the circumcenter of the projection of this 3D triangle. This guarantees the improvement of the mesh quality. Our approach is closely related to the problem of interpolation of bivariate functions: the input is a set of point S in the plane, along with a real-valued elevation $f(x,y)$ at each point $(x,y) \in S$. Any two dimensional triangulation $\mathcal{T}$ of the input points introduce a piecewise-linear function $f_{\mathcal{T}}$ defined on the region $\mathbb{R}$ bounded by the convex hull of S. For each point D, $f_{\mathcal{T}}(D)$ is the weighted average of the elevations at the vertices, A, B,and C of the triangle ABC in $\mathcal{T}$ that contains D. Writing D as $c_1A+c_2B+c_3C$ with $c_1+c_2+c_3=1$ and $c_1, c_2, c_3 \leq 0$. we have $f_{\mathcal{T}}(D) = c_1f(A) + c_2f(B) + c_3f(C)$. We say that $f_{\mathcal{T}}$ interpolates S. In interpolation of bivariate functions of points set data with elevation, Rippa [9] proved the following theorem.

Theorem 4.1 *Let $f_{\mathcal{T}}$ be a piecewise-linear function interpolating S induced by a triangulation $\mathcal{T}$. The piecewise-linear function $f_{\mathcal{DT}}$ induced by the Delaunay triangulation satisfies:*

$$\int\int_{\mathbb{R}^2} |\nabla f_{\mathcal{DT}})|^2 \, dxdy \leq \int\int_{\mathbb{R}^2} |\nabla f_{\mathcal{T}}|^2 \, dxdy \tag{3}$$

The proof of Rippa's theorem is an intricate calculation showing that the flip procudure can't increase the intergral [9]. Hence the CDT is also an **optimal interpolating surface**. Therefore, it guarantees that, in case of CDT, it is optimal for us to intruduce Steiner points into the mesh in the sense of the $\mathcal{L}_2$ norm of the gradient of the interpolant,

$$\mathcal{L}_2(f) = (\int\int_{\mathbb{R}^2} |\nabla f(x,y)|^2 \, dxdy)^{\frac{1}{2}} \tag{4}$$

From Rippa's theorem, we can see that, among all of the linear interpolations of the given points set induced by triangulations, if the domain of the function is constrained inside the polygons determined by the known points, the linear interpolation induced by the CDT has the minimum bending energy.

5 Experimental results

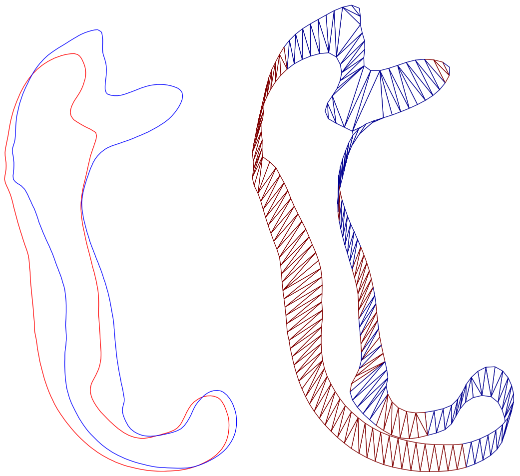

Figure 5: Adjacent contours and corresponding mesh without inserting Steiner points.

From Figures 6 and 7, we can see that the mesh quality is dramatically improved by introducing Steiner points. In Figure 7, for the mesh without Steiner points there are over 17% triangles who have angles less than 30 degree; but for the mesh with Steiner points, there are less than 1% triangles who have angles less than 30 degree. Figure 8 shows that we can get any detailed mesh by introducing enough Steiner points.

6 Conclusion

We introduced a new algorithm to build triangular meshes from 3D volumetric data based on modified polygon boolean operations and Constrained Delaunay Trianulation. The resulting mesh was shown to be the optimal one for surface interpolation. The algorithm can automatically resample the

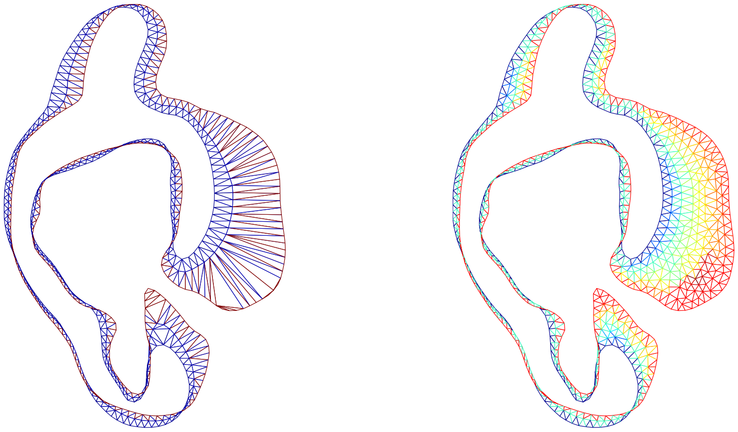

Figure 6: Top: Mesh without Steiner points; Bottom: Mesh with Steiner points

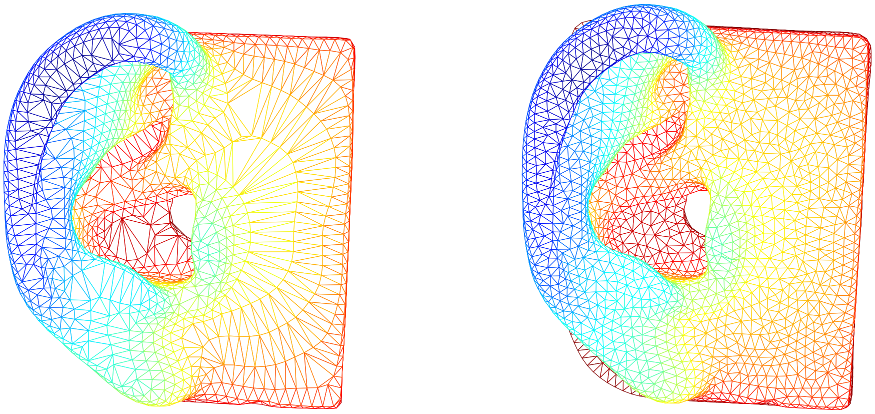

Figure 7: Pinna meshes. Top: mesh without Steiner points, many other surface reconstruction algorithms produce the similar results before remeshing. Bottom: mesh with Steiner points.

contours and insert Steiner points into the mesh to optimize the mesh quality and reach the desired resolution.

References

[1] A. C. Kak and Malcolm Slaney, Principles of Computerized Tomographic Imaging, *IEEE Press*, 1988.

[2] Sunder S. Rajan. MRI: A Conceptual Overview. *Springer Verlag*,January 1998.

[3] P. W. Hawkes, U. ValDre, Biophysical Electron Microscopy. *Academic Press*, 1990.

[4] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM J.Res.Dev 19*, 2-11, January 1975.

[5] H. Fuchs, Z. M. Kedem, S. P. Uselton. Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, **20**:693-702, 1977.

[6] James V. Miller, David E. Breen, William E. Lorensen, Robert M . O'Bara, Michael J. Wozny Geometrically Deformed Models: A methods for extracting Closed Geometric Models from Volume Data. *Computer Graphics*, **25**, 217-226, 1991.

[7] David Meyers, Shelley Skinner, Kenneth Sloan. Surface from Contours. *ACM Transactions on Graphics*, **11**(3):228-258, 1992.

[8] C. Gitlin, J. O'Rourke, V. Subramanian. On reconstructing polyhedra from parallel slices. *Technical Report 025, Department of Computer Science, Smith College,Northampton, MA*, 1993.

[9] S. Rippa. Minimal roughness property of Delaunay triangulation. *CAGD*, **7**:489-497,1990.

[10] S. Rippa, B. Schiff. Minimum Energy Triangulations for Elliptic Problems. *Computer Method in Applied Mechanics and Engineering*(84):257-274,1990.

[11] Ye Duan, Hong Qin. A Novel Modeling Algorithm for Shape Recovery of Unknown Topology. *Proceedings of The Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, 2001.

[12] Jean-Daniel Boissonnat, Bernhard Geiger. Three Dimensional Reconstrunction of Complex Shapes Based On the Delaunary Triangulation. *INRIA Tech Report 1697*, 1992.

[13] B. Zalik, M. Gombosi and D. Podgorelec. A Quick Intersection Algorithm for Arbitrary Polygons. *SCCG98 Conf. on Comput. Graphics and it's Applicat.*, 195-204, 1998.

[14] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. *First Workshop on Applied Computational Geometry (Philadelphia, Pennsylvania) 124-133, ACM*, 1996.

[15] Jim Ruppert. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms* **18**(3):548-585, 1995.

[16] Gill Barequet, Micha Sharir. Piecewise-linear Interpolation between Polygonal Slices. *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, 93–102, 1994.

[17] W. E. Lorensen, H. E. Cline Marching Cubes: A High Resolution 3D surface reconstruction algorithm. *Computer Graphics* **21**, 163-169, 1987.

[18] L. De Floriani, B. Falcidieno, C. Pienovi. A Delaunay-Based Method for Surface Approximation. *EuroGraphics 83*, 333-350, 1983.

[19] Marshall Bern, David Eppstein. Mesh Generation and Optimal Triangulation. *Computing in Euclidean Geometry Eds. D.Z. Du and F.K. Hwang, World Scientific Publishing Co.*, 23-90, 1992.

[20] Greg Turk, James F. O'Brien. Shape transformation using variational implicit functions. *International Conference on Computer Graphics and Interactive Techniques*, 335 - 342, 1999.

[21] G. Cong, B. Parvin. Robust and Efficient Surface Reconstruction from Contours *Visual Computers*, May, 2001.

[22] A. B. Ekoule, F. C. Peyrin, C. C. Odet. A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours. *ACM Transactions on Graphics* **10**(2):182-199, April 1991.

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

[24] 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.

[25] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. *Computer Graphics (SIGGRAPH 2001 proceedings)*, 67-76, August 2001.

[26] Heath, M., Sarkar, S., Sanocki, T., and Bowyer, K. Comparison of edge detectors: a methodology and initial study, *Computer Vision and Pattern Recognition '96, San Francisco*, June 1996.

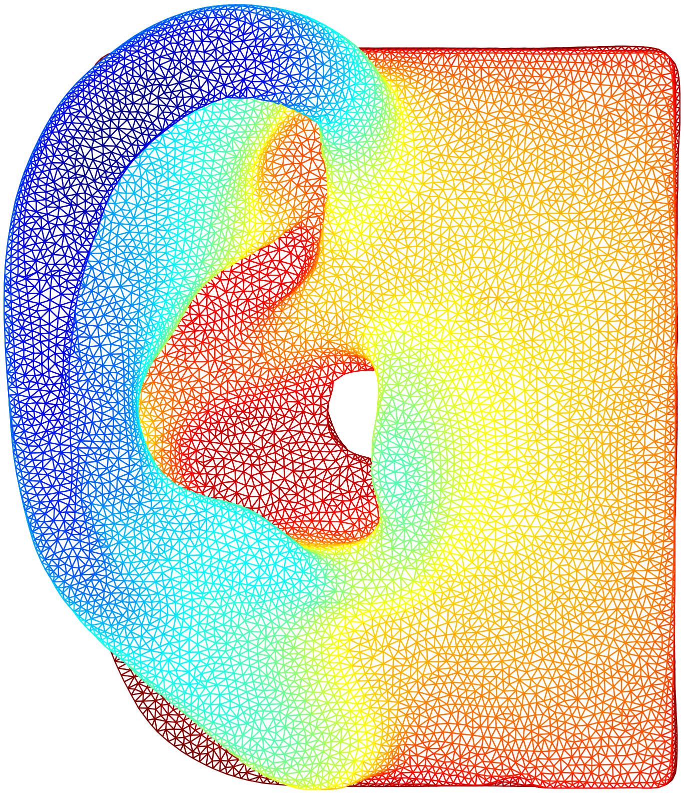

Figure 8: A detailed pinna mesh