Adaptive Multiresolution and Quality 3D Meshing from Imaging Data

Yongjie Zhang* Chandrajit Bajaj† Bong-Soo Sohn‡

Texas Institute for Computational and Applied Mathematics
Department of Computer Sciences
The University of Texas at Austin

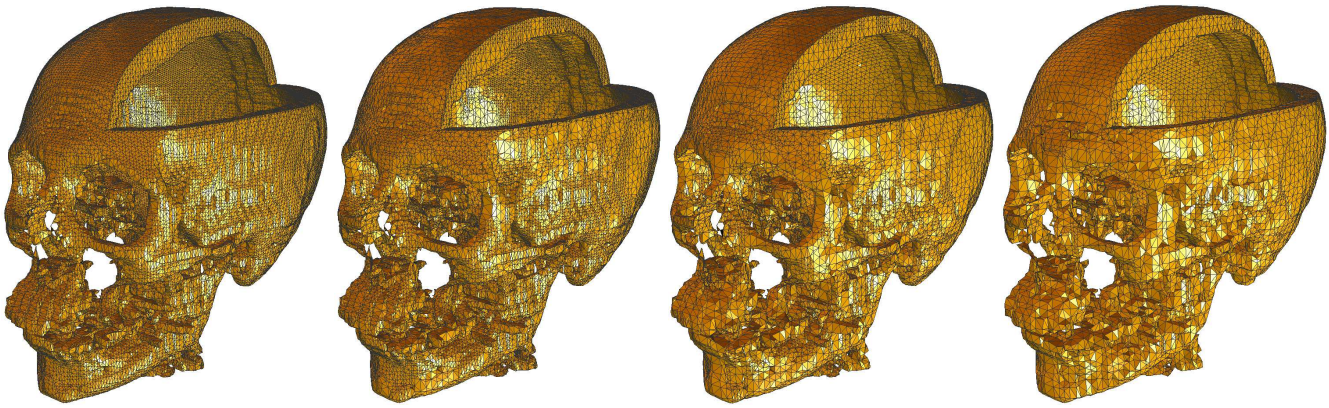

Figure 1: A series of adaptive tetrahedral meshes with different error tolerance are extracted from UNC Head (CT). Isovalues α_{in} = 1000.0, α_{out} = 120.0; error tolerance ε_{in} = 0.0001, (ε_{out}, the number of elements) = leftmost: (0.0001, 715892), middle left: (2.627, 579834), middle right: (4.817, 365215), rightmost: (9.999, 166271).

Abstract

This paper presents an algorithm to extract adaptive and quality 3D meshes directly from volumetric imaging data - primarily Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). The extracted tetrahedral and hexahedral meshes are extensively used in the Finite Element Method (FEM). Our comprehensive approach combines bilateral and anisotropic (feature specific) diffusion filtering, with contour spectrum based, relevant isosurface and interval volume selection. Then, a top-down octree subdivision coupled with the dual contouring method is used to rapidly extract adaptive and multiresolution 3D finite element (tetrahedral and hexahedral) meshes from volumetric imaging data. The main contributions are extending the dual contouring method to interval volume tetrahedralization and hexahedralization with curvilinear feature sensitive adaptation. Compared to other tetrahedral extraction methods from interval volumes (Marching Cubes and Marching Tetrahedra), our method generates high quality adaptive multiresolution 3D meshes without introducing any hanging nodes. Our method has the properties of crack prevention, feature preservation and feature sensitivity.

CR Categories: I.3.5 [Computation Geometry and Object Modeling]: CSG—Curve, surface, solid and object representations

Keywords: 3D meshes, adaptive, multiresolution, high quality, feature preservation, feature sensitive, crack prevention, hanging nodes

1 Introduction

The development of finite element simulations in medicine, molecular biology, engineering and geosciences has increased the need for high quality finite element meshes. Although there has been tremendous progresses in the area of surface reconstruction and 3D geometric modeling, it still remains a challenging process to generate 3D geometric models directly from imaging data, such as CT, MRI and signed distance function data (SDF). The image data can be represented as $V = \{F(i,j,k)|i,j,k$ are indices of x,y,z coordinates in a rectilinear grid$\}$. V is the volume containing function values $F(i,j,k)$ at the indices i,j,k.

Figure 2: Different Topology for Inner and Outer Surfaces

For accurate and efficient FEM calculations, it is important to have accurate and high quality models, minimize the number of elements and preserve features. The studied object may have complicated topologies. Figure 2 shows an interval volume between two isosurfaces in a SDF volumetric data of a knee. The two surfaces have the same topology in the left picture, while the topology of the inner surface may be different from the topology of the outer one (the right picture). In this paper, we present a comprehensive ap-

*jessica@ticam.utexas.edu
†bajaj@cs.utexas.edu
‡bongbong@cs.utexas.edu

proach to extract tetrahedral and hexahedral meshes directly from imaging data.

$$S_F(c) = \{(x,y,z) : F(x,y,z) = c\} \quad (1)$$
$$I_F(\alpha_1,\alpha_2) = \{(x,y,z) : \alpha_1 < F(x,y,z) < \alpha_2\} \quad (2)$$

Given volumetric imaging data and two isovalues α_1, α_2 ($\alpha_1 < \alpha_2$), each of which corresponds to an isosurface (Equation (1)). The algorithm to extract tetrahedral/hexahedral meshes from the interval volume, I_F, between the two isosurfaces is as follows:

1. Smoothing – Bilateral prefiltering and anisotropic diffusion.
2. Contour Spectrum based isosurface selection.
3. Adaptive multiresolution 3D meshing (tetrahedralization and hexahedralization) with feature preservation.
4. Quality improvement

Noise often exists in imaging data, especially in CT and MRI data. In order to reduce noise, the bilateral prefiltering and anisotropic diffusion methods [Bajaj et al. 2002] are used to smooth the volumetric data. Accurate gradient estimation can also be obtained. The Contour Spectrum [Bajaj et al. 1997] provides quantitative metrics of a volume to help us select two suitable isovalues for the interval volume.

In this paper, we extend the idea of dual contouring to interval volume tetrahedralization and hexahedralization from volumetric Hermite data (position and normal information). Dual Contouring [Ju et al. 2002] analyzes those edges that have endpoints which lie on different sides of the isosurface, called *sign change edge*. Each edge is shared by four (uniform case) or three (adaptive case) cells, and one minimizer is calculated for each of them by minimizing a predefined Quadratic Error Function (QEF) [Garland and Heckbert 1998].

$$QEF[x] = \sum_i (n_i \cdot (x - p_i))^2 \quad (3)$$

where p_i, n_i represent the position and unit normal vectors of the intersection point respectively. For each sign change edge, a quad or a triangle is constructed by connecting the minimizers. These quads and triangles provide an approximation of the isosurface.

Each sign change edge belongs to a boundary cell. In our tetrahedral mesh extracting process, we give a systematic way to tetrahedralize the volume in the boundary cell at the same time as isosurface extraction. For uniform grids, it is easy to deal with the interior cells. We only need to decompose each cell into five tetrahedra in a certain way. For the adaptive case, it is more complicated. In order to avoid introducing hanging nodes, which are strictly prohibited in finite element meshes, we design an algorithm to tetrahedralize the interior cell depending on the resolution levels of all its neighbors. As a by product, the uniform hexahedral mesh extraction algorithm is simpler. We analyze each interior vertex (a grid point inside the interval volume) which is shared by eight cells. One minimizer is calculated for each of them, and those eight minimizers construct a hexahedron.

In Dual Contouring, QEF is used for isosurface extraction and sharp features can be preserved. But what is a feature? How to identify features such as sharp edges and facial features (like nose, eyes, mouth and ears)? This paper introduces a different error function to identify those features sensitively.

The tetrahedral or hexahedral mesh extracted from volume data can not be used for finite element calculation directly, since some elements have bad quality. In order to satisfy the finite element requirements, the edge contraction method is used to improve the mesh quality.

The remainder of this paper is organized as the following: Section 2 summarizes the previous related tetrahedral and hexahedral extraction work; Section 3 introduces the outline of our comprehensive 3D mesh extracting method, and reviews the two preprocesses – noise smoothing (bilateral prefiltering and anisotropic diffusion) and isosurface selection (contour spectrum and contour tree). Section 4 explains the detailed algorithm of how to extract tetrahedra from the interval volume and how to hexahedralize the interval volume; Section 5 talks about the feature sensitive error function. Section 6 uses the edge contraction method to improve the mesh quality. Section 7 shows some results by applying our algorithm. The final section presents our conclusion.

2 Previous Work

In the last twenty years, the techniques of CT, MRI and ultrasound imaging (UI) have developed rapidly. Computer visualization, and engineering calculation (finite/boundary element analysis) require certain kinds of mesh extracted from these scanned volume data.

Anisotropic Diffusion The isotropic diffusion method can remove noise, but blurs features such as edges and corners. In order to preserve features during the process of noise smoothing, anisotropic diffusion [Weickert 1998] was proposed by introducing a diffusion tensor. Generally, a Gaussian filter is used to calculate the anisotropic diffusion tensor before smoothing, but it also blurs features. Bilateral filtering [Tomasi and Manduchi 1998], which is a nonlinear filter combining domain and range filtering, was introduced to solve this problem. Anisotropic diffusion can be used for fairing out noise both in surface meshes and functions defined on the surface [Bajaj and Xu 2002], [Clarenz et al. 2000].

Progressive Multiresolution Isosurface Extraction The predominant algorithm for isosurface extraction from volume data is Marching Cubes (MC) [Lorensen and Cline 1987], which computes a local triangulation within each cube to approximate the isosurface by using a case table of edge intersections. Furthermore, the asymptotic decider [Nielson and Hamann 1991] was proposed to avoid ambiguities existing in MC. For efficient isosurface extraction, [Bajaj et al. 1996b] starts from seed cells and traces the rest of the isosurface components by contour propagation.

Meshes extracted from uniform grids are usually dense, and adaptive meshes are preferable. When the adjacent cubes have different resolution levels, the cracking problem will happen. To keep the face compatibility, the gravity center of the coarser triangle is inserted, and a fan of triangles are used to approximate the isosurface [Westermann et al. 1999]. The chain-gang algorithm [Laramee and Bergeron 2002] was presented for isosurface rendering of super adaptive resolution (SAR) and resolves discontinuities in SAR data sets. Progressive multiresolution representation and recursive subdivision are combined effectively, and isosurfaces are constructed and smoothed by applying the edge bisection method [Pascucci and Bajaj 2000]. A surface wave-front propagation technique [Wood et al. 2000] is used to generate multiresolution meshes with good aspect ratio. By combining SurfaceNets [Gibson 1998] and the extended Marching Cubes algorithm [Kobbelt et al. 2001], octree based Dual Contouring [Ju et al. 2002] can generate adaptive multiresolution isosurfaces with good aspect ratio and preserve sharp features.

Quality and Feature Preserving Isosurface MC can not detect sharp features of the extracted isosurface, and severe alias artifacts appear. The enhanced distance field representation and the extended MC algorithm [Kobbelt et al. 2001] were introduced to extract feature sensitive isosurfaces from volume data. In order to improve the representation of the surface inside each cell, a trilinear function is used to modify the MC algorithm by identifying a small number of key points inside the cell that are critical to the surface definition [Lopes and Brodlie 2000].

Elements in the extracted mesh often have bad aspect ratio. These elements then can not be used for finite element calculations. The grid snapping method reduces the number of elements in an approximated isocontour and also improves the aspect ratio

of the elements [Moore 1992]. [Bern et al. 1990] studied how to generate triangular meshes with bounded aspect ratios from a planar point set. [Mitchell and Vavasis 2000] proposed an algorithm, called QMG, to triangulate a d-dimensional region with a bounded aspect ratio.

Quality Tetrahedral Mesh MC is extended to extract tetrahedral meshes between two isosurfaces directly from volume data, and a Branch-on-Need Octree is used as an auxiliary data structure to accelerate the extraction process [Fujishiro et al. 1996]. A different and systematic algorithm, Marching Tetrahedra (MT), was proposed for interval volume tetrahedralization [Nielson and Sung 1997]. A multiresolution framework [Zhou et al. 1997] was generated by combining recursive subdivision and edge-bisection methods. Since many 3D objects are sampled in terms of slices, Bajaj et. al introduced an approach to construct triangular surface meshes from the slice data [Bajaj et al. 1996a], and tetrahedralize the solid region bounded by planar contours and the surface mesh [Bajaj et al. 1999].

Poor quality tetrahedra called slivers are notoriously common in 3D Delaunay triangulations. Sliver exudation [Cheng et al. 2000] is used to eliminate those slivers. A deterministic algorithm [Cheng and Dey 2002] was presented for generating a weighted Delaunay mesh with no poor quality tetrahedra including slivers. Shewchuk [Shewchuk 2002a] solved the problem of enforcing boundary conformity by constrained Delaunay triangulation (CDT). Delaunay refinement [Shewchuk 1998], edge removal and multi-face removal optimization algorithm [Shewchuk 2002b] are used to improve the tetrahedral quality. Shewchuk [Shewchuk 2002c] gives some valuable conclusions on quality measures for finite element method.

Hexahedral Mesh Generation Hexahedral Mesh generation is a challenging problem. Eppstein [Eppstein 1996] starts from a tetrahedral mesh to decompose each tetrahedral into four hexahedra. Although this method avoids many difficulties, it increases the number of elements. Whisker Weaving [Folwell and Mitchell 1998] is an advancing front algorithm for hexahedral mesh construction, which is based on a global interpretation of the geometric dual of an all-hexahedral mesh. A trivariate subdivision scheme [Bajaj et al. 2001], consisting of a simple split and average algorithm, is described for hexahedral meshes.

3 Volumetric Imaging Pre-processing

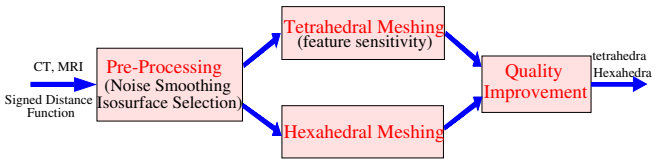

Figure 3: The Overview of the Comprehensive Method

Our comprehensive tetrahedral and hexahedral meshing method is displayed in Figure 3. We first use the anisotropic diffusion method coupled with bilateral prefiltering to remove noise from imaging data. Depending on the application, suitable isosurfaces are selected for the interval volume by using the contour spectrum and the contour tree. We then begin to extract tetrahedral or hexahedral meshes from the interval volume. In the process of 3D meshing, a feature sensitive error function is introduced to reduce the number of elements while preserving features. Finally, the edge contraction method is used to improve the quality of the extracted mesh.

Since noise in imaging data influences the accuracy of the extracted meshes, it is important to remove it before the mesh extracting process. We use the anisotropic diffusion method [Bajaj et al. 2002] to smooth noise. In order to obtain more accurate computation of curvature and gradient for anisotropic diffusion tensor, the bilateral prefiltering combining the domain and range filtering together is chosen instead of Gaussian filtering because it can preserve features such as edges and corners.

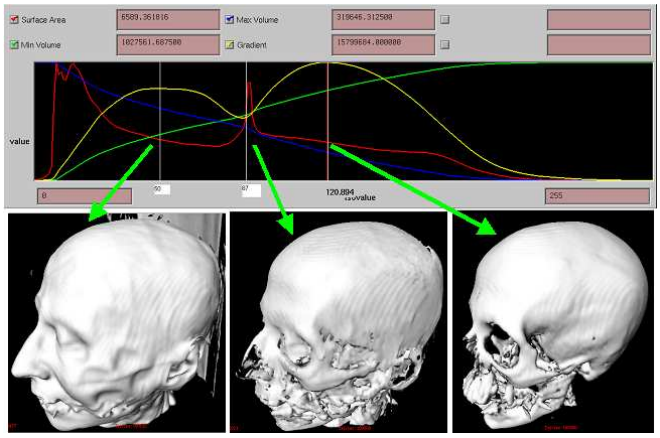

Figure 4: The Contour Spectrum of the UNC Human Head

Mesh extraction from imaging data requires selection of suitable boundary isosurfaces. We use a user interface called Contour Spectrum [Bajaj et al. 1997], to find isosurfaces of interest. The Contour Spectrum computes quantitative properties such as surface area, volume, and gradient integral of contours in real time, and helps to choose suitable isosurfaces by showing the related spectrum in a 2D plane. For instance, we can obtain isosurfaces of a skin and a bone in CT scanned human head data (Figure 4) by taking isovalues having the local maximum of the gradient integral.

In the case of interval volumes, the topology of inner and outer isosurfaces may need to be controlled depending on the application. A contour tree [Carr et al. 2003] can be used to capture the topological information on each isosurface and help choose isosurfaces with desirable topology. For example, we can choose the inner and outer isosurfaces with exactly same topology by taking isovalues lying on the same edges in a contour tree.

4 3D Mesh Extraction

In this section, our goal is to tetrahedralize or hexahedralize the interval volume between two isosurfaces by using an octree-based data structure. We describe in detail how to extract adaptive multiresolution tetrahedral meshes from volume data. First, we discuss triangulation in 2D problems, then we extend it to 3D tetrahedralization. A hexahedral mesh generation algorithm is presented at the end of this section.

4.1 Uniform Tetrahedral Extraction

For isosurface extraction, we only need to analyze boundary cells – those cells that contain sign change edges, or those cells that contain the isosurface. There are four neighbor cubes which share the same sign change edge. Uniform Dual Contouring generates one minimal vertex for each neighbor cube by minimizing the QEF, and then connects them to generate a quad. By marching all the sign change edges, the isosurface is obtained.

For tetrahedral mesh extraction, cells inside the interval volume should also be set as leaves besides the boundary cells.

DEFINITION 1 (Sign Change Edge): A sign change edge is an edge whose one vertex lies inside the interval volume (we call it the interior vertex of this sign change edge), while the other vertex lies outside.

DEFINITION 2 (Interior Edge in Boundary Cell): In a boundary cell, those edges with both vertices lying inside the interval volume are called interior edges.

DEFINITION 3 (Interior Cell): different from the boundary cell, all the eight vertices of an interior cell lie interior to the interval volume.

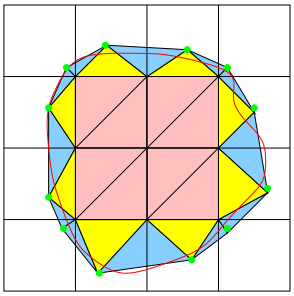

Figure 5: Uniform Triangulation - the red curve represents the isocontour, and green points represent minimizers.

4.1.1 Uniform 2D Triangulation

Figure 5 is a uniform triangulation example of the area interior to the isocontour in two dimensions. There are three different cases which need to be dealt with separately.

1. Sign change edge – find the minimizers of the two cells which share the edge, then the two minimizers and the interior vertex of the edge construct a triangle (blue triangles).

2. Interior edge in boundary cell – find the QEF minimizer of the boundary cell, then the minimizer and this interior edge construct a triangle (yellow triangles).

3. Interior cell – decompose each interior cell into two triangles (pink triangles).

4.1.2 Uniform 3D Tetrahedralization

DEFINITION 4 (Interior Face in Boundary Cell) : in the boundary cell, those faces with all four vertices lying inside the interval volume are called interior faces.

Compared to 2D triangulation, three dimensional tetrahedral meshing is more complicated.

1. Sign change edge – decompose the quad into two triangles, then each triangle and the interior vertex of this edge construct a tetrahedron. In Figure 6(a), the red line represents the sign change edge, and two blue tetrahedra are constructed.

2. Interior edge in boundary cell – find the QEF minimizers of the boundary cell and its boundary neighbor cells, then two adjacent minimizers and the interior edge construct a tetrahedron. In Figure 6(b)(c), the red cube edge represents the interior edge. (b) gives four minimizers to construct four edges, each of which construct a tetrahedron with the interior edge, so totally four tetrahedra are constructed. While (c) assumes the cell below this boundary cell is interior the interval volume, so there is no minimizer for it. Therefore we obtain three minimizers, and only two tetrahedra are constructed.

3. Interior face in boundary cell – find the QEF minimizer of the boundary cell, then the interior face and the minimizer construct a pyramid, which can be decomposed into two tetrahedra (Figure 6(f)). Figure 6(d)(e)(f) give a sequence how to generate tetrahedra when there is only one interior face in the boundary cell. (d) analyzes four sign change edges, (e) deals with four interior edges and (f) fills the gap.

4. Interior cell – decompose the interior cube into five tetrahedra. There are two different decomposition ways (Figure 6(g)(h)). For two adjacent cells, we choose a different decomposition method to avoid the diagonal choosing conflict problem.

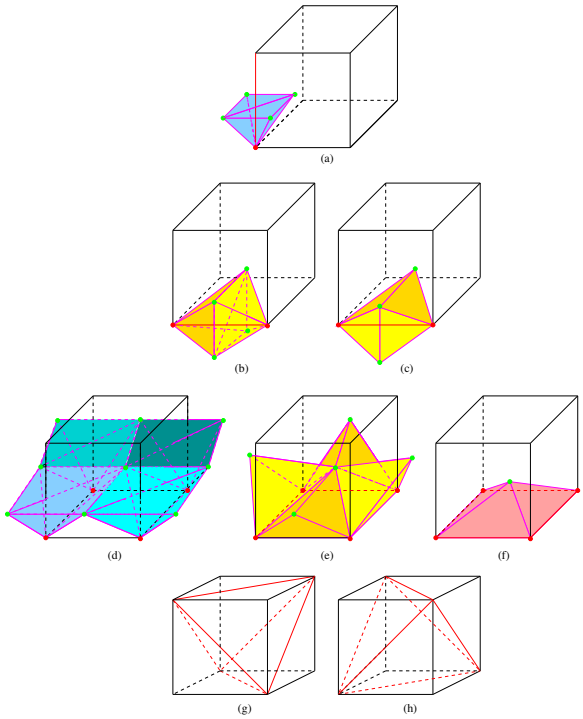

Figure 6: Case Table of Uniform Tetrahedralization - the red vertex means it lies interior to the interval volume, otherwise, it is outside. Green points represent minimizers. (a) - Sign Change Edge; (b)(c) - Interior Edge in Boundary Cell; (d)(e)(f) - Interior Face in Boundary Cell; (g)(h) - Interior Cell.

If two isosurfaces pass across the same sign change edge, we need analyze the sign change edge twice. In 2D case (Figure 7, Left picture) two minimizers are obtained for the inner surface, and similarly two minimizers are calculated for the outer surface. They construct a quad, which can be decomposed into two triangles (yellow ones). For 3D case (Figure 7, Right picture), a hexahedron is built between the two surfaces for the sign change edge. The hexahedron can be split into five tetrahedra.

Two different isosurfaces can not intersect with each other, since one point can not have two isovalues. However, the two quads approximating the two isosurfaces may intersect because of bad gradient information. This case needs to be detected carefully, otherwise the extracted tetrahedra will intersect with each other. This can be solved by splitting the cell into eight cubes in the octree data structure, and then analyzing those cells separately.

4.2 Adaptive Tetrahedral Extraction

Uniform tetrahedralization usually gives an over-sampled mesh. Adaptive tetrahedral meshing is a good and effective way to reduce the number of elements while preserving the accuracy requirement.

First, we split the volume data by using the octree data structure to obtain denser cells along the boundary, and coarser cells inside the interval volume. The QEF value is calculated for each octree cell, and a much more efficient octree is built by comparing the QEF value with a given error tolerance ε and using the bottom-up

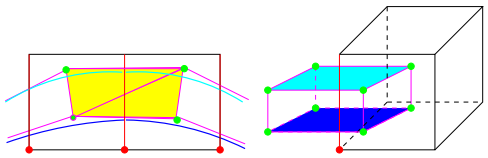

Figure 7: Sign Change Edge Passed Across by Two Isosurfaces - Left (2D) : the cyan and blue curves represent the two isocontour; Right (3D): the cyan and blue quads approximate the two isosurfaces. The red edges are sign change edges and green points represent minimizers.

algorithm. Leaves of the octree have different resolution levels. The next step is to analyze each leaf.

Each leaf cell may have neighbors at different levels. An edge in a leaf cell may be divided into several edges in its neighbor cells. Therefore it is important to decide which edge should be analyzed. The Dual Contouring isosurface method gives a good rule to follow – we always choose the minimal edges. Minimal edges are those edges of leaf cubes that do not properly contain an edge of a neighboring leaf.

Similar to uniform tetrahedral mesh extraction, we need to analyze the sign change edge, the interior edge and the interior face in the boundary cell, and the interior cell. When we analyze boundary cells, only minimal edges and minimal faces are analyzed. Compared to the uniform case, the only difference is in how to decompose the interior cell into tetrahedra without hanging nodes.

DEFINITION 5 (Hanging Node) : a hanging node is one that is attached to the corner of one triangle but does not attach to the corners of the adjacent triangles. Generally, a hanging node is a point that is a vertex for some elements (e.g., triangle, quad, tetra, hexa), but it is not for its other neighbor elements that share it. It lies on one edge or one face of its neighbors, for example, a T-Vertex.

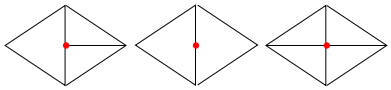

Figure 8: Hanging Node Removal - the red point is a hanging node. Left - T-vertex; Middle - merging two triangles; Right - splitting method.

Figure 8 shows two methods to remove hanging nodes - splitting and merging. In the T-vertex example (left picture), only the right triangle has hanging nodes removal problem if we use the merging method; while only the left one has hanging nodes removal problem if we tend to split the mesh. In order to maintain accuracy, we adopt the splitting method in our algorithm.

LEMMA: Only the interior cell has the hanging nodes removal problem if the splitting method is adopted.

Proof: All the leaf cells can be divided into two groups: the boundary cell and the interior cell.

1. interior cell – Since its neighbor cells may have higher resolution level, hanging nodes are unavoidable.

2. boundary cell – There are two rules for the sign change edge, the interior edge and the interior face. The two rules guarantee that there are no hanging nodes removal problem for the boundary cell if only the splitting method is chosen.

   - Minimal Edge/Face Rule - only minimumal edges/faces are chosen. This rule keeps the analyzed edges/faces owning the highest resolution level compared with its neighbors.
   - Only one minimizer is generated for each leaf cell.

4.2.1 Adaptive 2D Triangulation

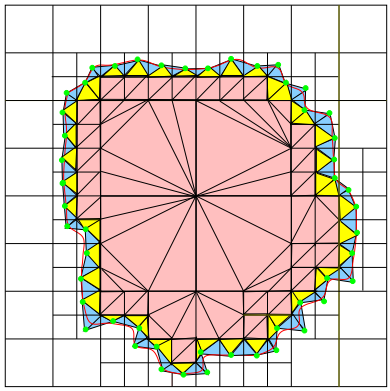

Figure 9: Adaptive Triangulation. The red curve represents the isocontour, green points represent minimizers.

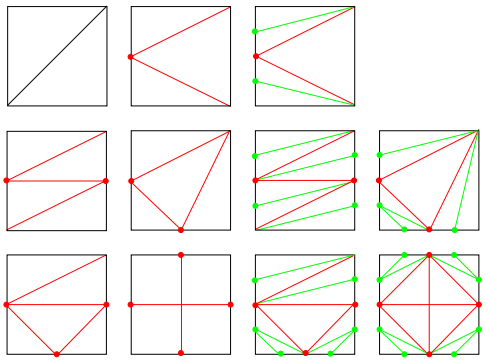

Figure 10: Case Table for Decomposing the Interior Cell into Triangles. Suppose the resolution level of this cell is κ, and middle points appear on the shared edges if its neighbors has higher level than κ. Red points and red lines mean its neighbors have level (κ+1); green points and green lines mean its neighbors have higher level than (κ+1).

Figure 9 gives an example of how to triangulate the interior area of an isocontour. Similarly, we need to analyze the following three problems:

1. Sign change edge – if the edge is minimal, deal with it as in the uniform case (blue triangles). Otherwise, skip it.

2. Interior edge in the boundary cell – if the edge is minimal, deal with it as in the uniform case (yellow triangles). Otherwise, skip it.

3. Interior cell – Figure 10 lists all the cases of how to decompose the interior cell into triangles.

Compared to the uniform case, the triangulation of interior cells is more complicated. All neighbors of an interior cell need to be checked because the neighbor cells are used to decide if there are any middle points on the shared edge. Suppose the resolution level of this cell is κ, we group into five cases according to the number of edges whose level is greater than κ. The i^{th} group means there are number i edges whose level is greater than κ, where i = 0, ..., 4. For each subdivided edge, it may be subdivided more than once, or the neighbor cell may have higher level than (κ+1). So we need to search all the middle points on this edge. Top-down or bottom-up algorithm can be used here to find the resolution level of its neighbors, and find out all the middle points on the edge. Figure 10 gives

all the cases of how to decompose the interior cell into triangles according to its neighbors' resolution levels. If all the four edges have already been subdivided, then we can use the recursion method to march each of the four smaller cells with the same algorithm. In this way, hanging nodes are removed effectively.

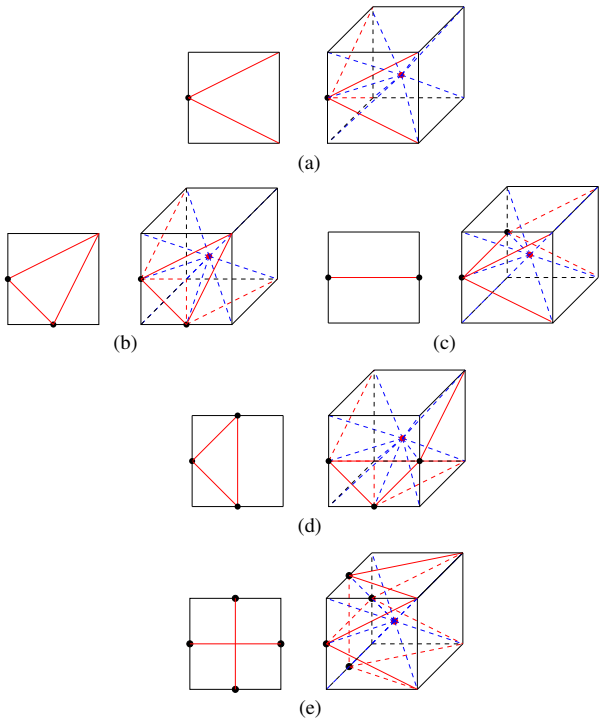

Figure 11: Case Table for Decomposing the Interior Cell into tetrahedra – in (a $\sim$ e), the left picture gives the triangulation format of one face according to Figure (10); the right one shows how to decompose the cell into tetrahedra without hanging nodes. (a) - one subdivided edge; (b)(c) - two subdivided edges; (d) - three subdivided edges; (e) - four subdivided edges.

4.2.2 Adaptive 3D Tetrahedralization

For three dimensional adaptive tetrahedralization, we use the same algorithm with the uniform case when we deal with the boundary cell.

1. Sign change edge – if the edge is minimal, deal with it as in the uniform case. Otherwise, skip it.

2. Interior edge in the boundary cell – if the edge is minimal, deal with it as in the uniform case. Otherwise, skip it.

3. Interior face in boundary cell – identify all the middle points on the four edges, and decompose the face into triangles by applying the same algorithm as in the adaptive 2D case, then calculate the minimizer of this cell, each triangle and this minimizer construct a tetrahedron.

4. Interior cell – decompose each face of the cube into triangles, just as how to deal with the interior cell for the adaptive 2D triangulation (Figure 10), then insert a Steiner point at the cell center. Each triangle and the Steiner point construct a tetrahedron. Figure 11 shows how to construct tetrahedra. Sometimes pyramids are constructed, in order to avoid the diagonal choosing conflict, we decide which diagonal is chosen to decompose one pyramid into two tetrahedra according to the odd-even property of the cell index.

By using the above algorithm, we extract tetrahedral meshes from volumetric imaging data successfully. Figure 12 (Upper Row) gives one example – the tetrahedral mesh of the human head model extracted from 65*65*65 volume data. The volume inside the skin isosurface is tetrahedralized.

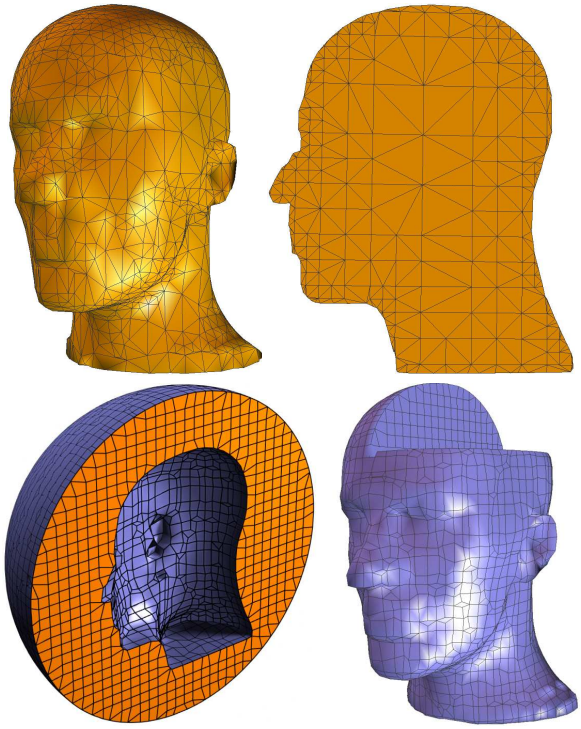

Figure 12: Tetrahedral/Hexahedral Meshes of the Human Head Model - The upper row shows an adaptive tetrahedral mesh; the lower row gives the hexahedral meshes, Left: hexahedralization of the volume inside the head, Right: hexahedralization of the volume between the human head and a sphere boundary.

4.3 Hexahedral Extraction

Finite element calculations sometimes require hexaderal meshes instead of tetrahedral meshes.

Each hexahedron has eight points. In the tetrahedralization process we deal with edges shared by at most four cells. This means that we can not get eight minimizers for each edge. But, each vertex is shared by eight cells, and we can calculate a minimizer for each of them. These eight minimizers can then be used to construct a hexahedron.

Figure 12 (lower row) shows two hexahedral meshes for the head model, which is used to solve electromagnetic problems.

5 Feature Sensitive Error Function

For a geometric object, a feature appears where the local shape of some area changes (the geometric gradient vector is nontrivial). A clear feature is generated by a rapid shape change (large gradient vector). For example, the gradient in the sharp edge area is large, so the sharp edge is a feature; For the human head model, in the areas where the nose, eyes, mouth and ears are located, the shape changes a lot locally and the gradient is nontrivial, so they are obvious features.

For efficiency and accuracy during calculations, finite element applications require the number of elements to be as small as possible, while preserving necessary features. For a given precision requirement, the uniform mesh is always over-sampled with unnecessary small elements. Adaptive meshes are therefore preferable.

For the adaptive mesh, an error function and an error tolerance ε are required, which set the criteria to identify where we should

select higher level (denser mesh) and where lower level (coarser mesh) should be chosen. In order to minimize the number of elements and preserve features at the same time, it is important to have a feature sensitive error function.

The Dual Contouring algorithm can preserve sharp features by using the QEF error function. Examples show that it is not sensitive to the gradient based features, for example, facial features, like the nose, eyes, mouth and ears of the human head model in Figure 14. We know that the local gradient vector influences the feature directly. Since we work with Hermite imaging data, we have not only the grid positions but also the information of the first derivatives. Therefore we choose a different error function related to the gradient information in order to identify features.

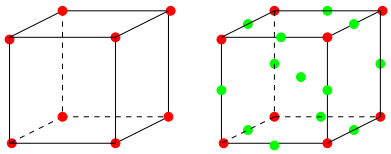

Figure 13: Sampling Points for Feature Sensitive Error Function - red points are eight vertices of the cell and green points are edge middle points. Left - Level (i); Right - Level (i+1).

In Figure 13, the left picture gives eight red vertices for resolution level (i), and the right picture gives thirteen green middle points besides those eight red vertices, which represents level (i+1). For level (i), the eight red vertices' function values are given, and a trilinear function is defined in Equation (4) , from which the function values of edge middle points (green ones) and face middle points (not shown in Figure 13) can be obtained. For level (i+1), the function values of all vertices and edge/face middle points are given. The error function is defined in Equation (5).

$$\begin{aligned} f^i(x,y,z) &= f_{000}(1-x)(1-y)(1-z) \\ &+ f_{001}(1-x)(1-y)z \\ &+ f_{010}(1-x)y(1-z) \\ &+ f_{011}(1-x)yz \\ &+ f_{100}x(1-y)(1-z) \\ &+ f_{101}x(1-y)z \\ &+ f_{110}xy(1-z) \\ &+ f_{111}xyz \end{aligned} \tag{4}$$

$$EDerror = \sum \frac{|f^{i+1} - f^i|}{|\nabla f^i|} \tag{5}$$

The two error functions are compared in Figure 14. It is obvious that our error function can preserve sharp edges, and is more sensitive to the areas where nose, eyes, mouth and ears are located on the human head model. In our error function, we use the gradient to normalize the difference of function values between two levels. Thus the error function can reflect the gradient information, or the features. This is why the error function identifies features sensitively.

6 Quality Improvement

The above 3D mesh extraction algorithm can tetrahedralize the interval volume, and the extracted meshes have better quality than meshes from other methods such as MC and MT. However, it can not guarantee that all the elements have good quality. For example, sliver triangles or tetrahedra exist. In this section, we focus on how to improve the quality of tetrahedral meshes.

In order to measure tetrahedra's quality, three quality parameters are borrowed from ABAQUS document (a FEM software).

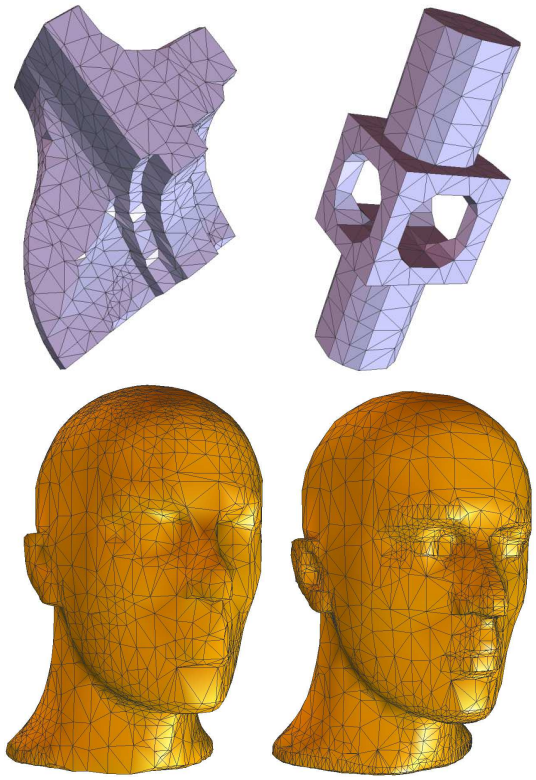

Figure 14: Upper Row – sharp edge features; Lower Row – facial features: comparison of QEF (left picture, 2952 triangles) and the euclidean distance error (EDerror) function (right picture, 2734 triangles).

- Tetrahedral Quality Measure = volume of tetrahedron / volume of equilateral tetrahedron with same circum-sphere radius (> 0.02)
- Min/Max Angles (Figure 16) – with minimum angle $\alpha > 10^\circ$ and maximum angle $\beta < 160^\circ$, where

$$\alpha_A = min(\alpha_1, \alpha_2, \alpha_3) \tag{6}$$
$$\beta_A = max(\alpha_1, \alpha_2, \alpha_3) \tag{7}$$
$$\alpha = min(\alpha_A, \alpha_B, \alpha_C, \alpha_D) \tag{8}$$
$$\beta = max(\beta_A, \beta_B, \beta_C, \beta_D) \tag{9}$$

  where, α_B, α_C, α_D, β_B, β_C and β_D have the similar definition as α_A and β_A.
- Right-hand-side principle

In the process of improving the mesh quality, edge contraction is a direct method to eliminate sliver tetrahedra. The mesh modification process can be divided into three steps. For each tetrahedron,

1. Calculate the three quality parameters.
2. If the tetrahedron's orientation is Left-hand-side, swap any two vertices' index number.
3. If Tetrahedral Quality Measure ≤ 0.02 or Min Angle $\leq 10^\circ$ or Max Angle $\geq 160^\circ$, e.g., in Fig. 16, $\alpha_A \leq 10^\circ$. Find the shortest edge among AB, AC and AD, then merge vertex A with the closest point.

If sliver tetrahedra still exist, repeat steps 1 – 3. Figure 17 shows an example.

Data Set	Type	Resolution	Number of Tetrahedra (Extraction Time (unit : ms))			
			(a)	(b)	(c)	(d)
UNC Head (Skull)	CT	$129 \times 129 \times 129$	715892 (12985)	579834 (10203)	365215 (7922)	166271 (3063)
UNC Head (Skin)	CT	$129 \times 129 \times 129$	–	935124 (17406)	545269 (10468)	69351 (312)
Poly	CT	$257 \times 257 \times 257$	–	276388 (5640)	63325 (1672)	14204 (672)
Head	SDF	$65 \times 65 \times 65$	143912 (2547)	76218 (1391)	40913 (766)	10696 (203)
Knee	SDF	$65 \times 65 \times 65$	70768 (1360)	94586 (1782)	93330 (1750)	72366 (1406)

Figure 15: Data Sets and Test Results. The CT data sets are re-sampled to fit into the octree representation. Rendering results for each case is shown in Figure 18, 19, 20, 21.

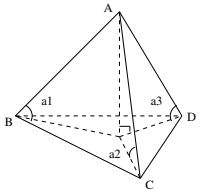

Figure 16: Definition of Min/Max Angles for Tetrahedra

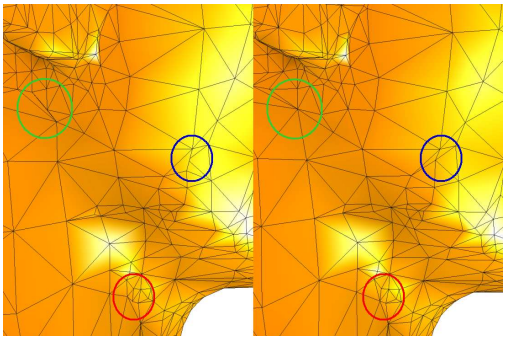

Figure 17: Quality Improvement - Left: no edge contraction, circles mark triangles with bad aspect ratio; Right: poor quality triangles disappear after edge contraction.

7 Results

We developed an interactive program for 3D mesh extraction and rendering from volumetric data sets. In the program, the error tolerance and the isovalues can be changed interactively. The results were computed on a PC equipped with a Pentium III 800 MHz processor and 1 GB main memory.

We tested our algorithm on volumetric data from CT scans, *the UNC Human Head* and *Poly (heart valve)*, and signed distance volumes generated from the polygonal surfaces of a human head and a knee.

Figure 15, 18, 19, 20, 21 provide information about data sets and test results. The results consist of the number of tetrahedra, extraction time, and corresponding images with respect to different isovalues and error tolerance. As a preprocessing, we calculate min/max values for each octree cell to visit only cells contributing to mesh extraction and to compute QEF values only in those cells at run time. Extraction time in the table includes octree traversal, QEF computation and actual mesh extraction, given isovalues and error tolerance values for inner and outer surfaces as run time parameters. If we fix isovalues, and change error tolerance interactively, the computed QEF is reused and thus the whole extraction process is accelerated.

To extract 3D meshes from the surface data, we computed signed distance function from the surface and performed mesh extraction. Figure 18 and 19 shows the resulting tetrahedral meshes extracted from two signed distance function data with the size of 65^3. The results from CT data are shown in Figure 1 (head skull), 20 (head skin) and 21 (heart valve). The number of elements in the extracted mesh is controlled by changing error tolerance. It is clear to see that adaptive tetrahedral meshes are extracted from the interval volume, and features are identified sensitively and preserved (Figure 18). In Figure 19, the sequence of images are generated by changing the isovalue of the inner isosurface. The topology of the inner isosurface can change arbitrarily.

8 Conclusion

We have presented an algorithm to extract adaptive and high quality 3D meshes directly from volumetric imaging data. By extending the dual contouring method described in [Ju et al. 2002], our method can generate 3D meshes with good properties such as multiresolution with no hanging nodes, sharp feature preservation and good aspect ratio. Using an error metric which is normalized by the function gradient, the resolution of the extracted mesh is adapted to the features sensitively. The resulting meshes are useful for efficient and accurate FEM calculations.

Acknowledgments

We thank Qiu Wu, and Anthony Thane for the use of their volume diffusion and rendering program, Jianguang Sun for the system management, Vinay Siddavanahalli for useful discussions, Shashank G. Khandelwal for proofreading, and UNC for providing access to the CT volume dataset of a human head. This work was supported in part by NSF grants ACI-9982297, CCR-9988357, a DOE-ASCI grant BD4485-MOID from LLNL/SNL and from grant UCSD 1018140 as part of NSF-NPACI, Interactive Environments Thrust.

References

BAJAJ, C., AND XU, G. 2002. Anisotropic diffusion of subdivision surfaces and functions on surfaces. *To appear in ACM Transactions on Graphics*.

BAJAJ, C., COYLE, E., AND LIN, K.-N. 1996. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Modeling and Image Processing 58*, 6, 524–543.

BAJAJ, C. L., PASCUCCI, V., AND SCHIKORE, D. 1996. Fast isocontouring for improved interactivity. In *Proceedings of the IEEE Symposium on Volume Visualization, ACM Press*, 39–46.

BAJAJ, C., PASCUCCI, V., AND SCHIKORE, D. 1997. The contour spectrum. In *Proceeding of IEEE Visualization 1997*, 167–174.

BAJAJ, C., COYLE, E., AND LIN, K.-N. 1999. Tetrahedral meshes from planar cross sections. *Computer Methods in Applied Mechanics and Engineering 179*, 31–52.

BAJAJ, C., WARREN, J., AND XU, G. 2001. A smooth subdivision scheme for hexahedral meshes. In *The Visual Computer, the special issue on subdivision.*

BAJAJ, C., WU, Q., AND XU, G. 2002. Level set based volume anisotropic diffusion. In *TICAM Technical Report, The Univ. of Texas at Austin*.

BERN, M. W., EPPSTEIN, D., AND GILBERT, J. R. 1990. Provably good mesh generation. In *IEEE Symposium on Foundations of Computer Science*, 231–241.

CARR, H., SNOEYINK, J., AND AXEN, U. 2003. Computing contour trees in all dimensions. *Computational Geometry : Theory and Applications 24*, 2, 75–94.

CHENG, S.-W., AND DEY, T. K. 2002. Quality meshing with weighted delaunay refinement. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms (SODA 2002)*, 137–146.

CHENG, S.-W., DEY, T. K., EDELSBRUNNER, H., FACELLO, M. A., AND TENG, S. 2000. Sliver exudation. *Proc. Journal of ACM 47*, 883–904.

CLARENZ, U., DIEWALD, U., AND RUMPF, M. 2000. Nonlinear anisotropic diffusion in surface processing. In *Proceedings of the 11th IEEE Visualization Conference (Vis) 2000*, 397–405.

EPPSTEIN, D. 1996. Linear complexity hexahedral mesh generation. In *Symposium on Computational Geometry*, 58–67.

FOLWELL, N., AND MITCHELL, S. 1998. Reliable whisker weaving via curve contraction. In *Proceedings, 7th International Meshing Roundtable, Sandia National Lab*, 365–378.

FUJISHIRO, I., MAEDA, Y., SATO, H., AND TAKESHIMA, Y. 1996. Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics 2*, 2, 144–155.

GARLAND, M., AND HECKBERT, P. S. 1998. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization '98*, D. Ebert, H. Hagen, and H. Rushmeier, Eds., 263–270.

GIBSON, S. F. 1998. Using distance maps for accurate surface representation in sampled volumes. In *IEEE Visualization '98*, 23–30.

JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. 2002. Dual contouring of hermite data. In *Proceedings of SIGGRAPH 2002*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM.

KOBBELT, L., BOTSCH, M., SCHWANECKE, U., AND SEIDEL, H. 2001. Feature sensitive surface extraction from volume data. In *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, 57–66.

LARAMEE, R., AND BERGERON, R. 2002. An isosurface continuity algorithm for super adaptive resolution data. *Computer Graphics International, Bradford, UK, July 1-5 2002. Computer Graphics Society. 5*.

LOPES, A., AND BRODLIE, K. 2000. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. In *IEEE Transactions on Visualization and Computer Graphics*.

LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH 1987*, Computer Graphics Proceedings, 163–169.

MEYERS, D. 1994. Multiresolution tiling. In *Proceedings of Graphics Interface*, 25–32.

MITCHELL, S. A., AND VAVASIS, S. A. 2000. Quality mesh generation in higher dimensions. *SIAM Journal on Computing 29*, 4, 1334–1370.

MOORE, D. 1992. Compact isocontours from sampled data. *Graphics Gems III*, 23–28.

NIELSON, G. M., AND HAMANN, B. 1991. The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proceedings of Visualization '91*, 83–90.

NIELSON, G. M., AND SUNG, J. 1997. Interval volume tetrahedrization. In *IEEE Visualization '97*, 221–228.

PASCUCCI, V., AND BAJAJ, C. 2000. Time critical adaptive refinement and smoothing. In *Proceedings of the ACM/IEEE Volume Visualization and Graphics Symposium 2000*, Salt lake City, Utah, 33–42.

SHEWCHUK, J. R. 1998. Tetrahedral mesh generation by delaunay refinement. In *Proceedings of the 4th Annual Symposium on Computational Geometry (Minneapolis, Minnesota), Association for Computational Machinery*, 86–95.

SHEWCHUK, J. R. 2002. Constrained delaunay tetrahedrizations and provably good boundary recovery. In *To appear in the 11th International Meshing Roundtable*.

SHEWCHUK, J. R. 2002. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. In *Unpublished manuscript*.

SHEWCHUK, J. R. 2002. What is a good linear element? interpolation, conditioning, and quality measures. In *To appear in the 11th International Meshing Roundtable*.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India*, 839–846.

WEICKERT, J. 1998. Anisotropic diffusion in image processing. B. G. Teubner Stuttagart.

WESTERMANN, R., KOBBELT, L., AND ERTL, T. 1999. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer 15*, 2, 100–111.

WOOD, Z., DESBRUN, M., SCHRODER, P., AND BREEN, D. 2000. Semi-regular mesh extraction from volumes. In *Visualization 2000 Conference Proceedings*, 275–282.

ZHOU, Y., CHEN, B., AND KAUFMAN, A. 1997. Multiresolution tetrahedral framework for visualizing regular volume data. In *Proceedings of the 8th IEEE Visualization '97 Conference*, 135–142.

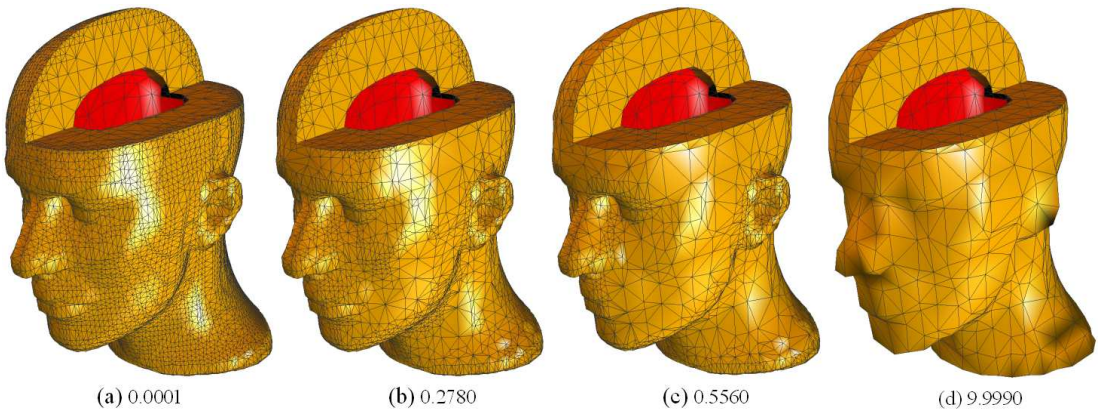

Figure 18: Head (SDF) – Isovalues α_{in} = -9,17464, α_{out} = 0.0001; error tolerances ε_{in} = 1.7, ε_{out} are listed below each picture

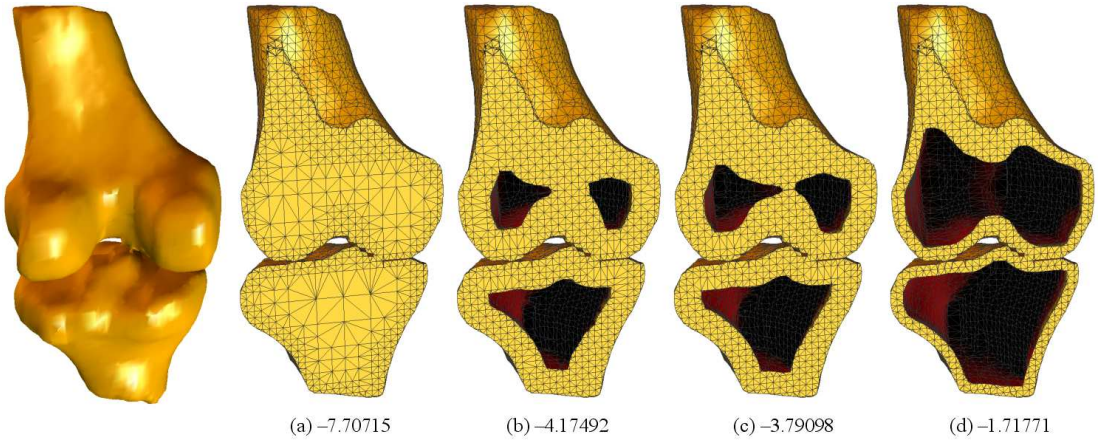

Figure 19: Knee (SDF) – Error tolerances $\varepsilon_{in} = \varepsilon_{out}$ = 0.0001; isovalues α_{out} = -0.02838, α_{in} are listed below each picture

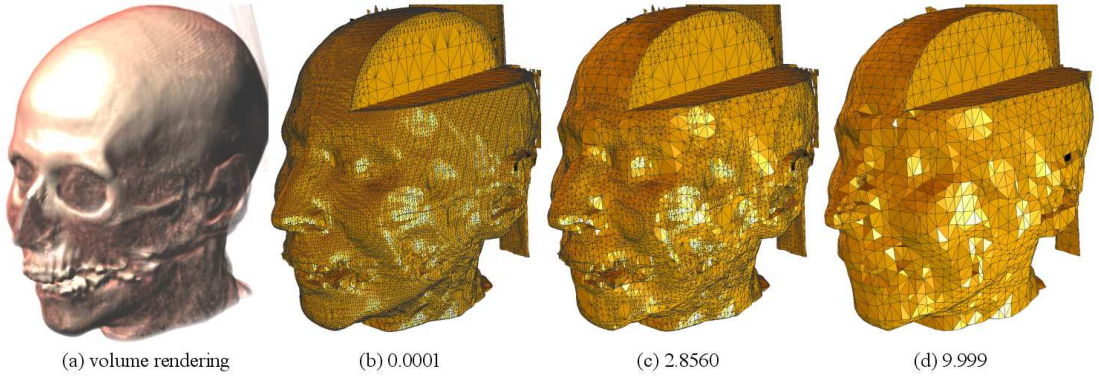

Figure 20: UNC Head Skin (CT) – Isovalues α_{in} = 1000.0, α_{out} = 50.0; error tolerances ε_{in} = 0.0001, ε_{out} are listed below each picture

Figure 21: Heart Valve (Poly) – Isovalues α_{in} = 1000.0, α_{out} = 75.0; error tolerances ε_{in} = 0.0001, ε_{out} are listed below each picture