Adaptive and Quality 3D Meshing from Imaging Data

Yongjie Zhang[*] Chandrajit Bajaj[†] Bong-Soo Sohn[‡]

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

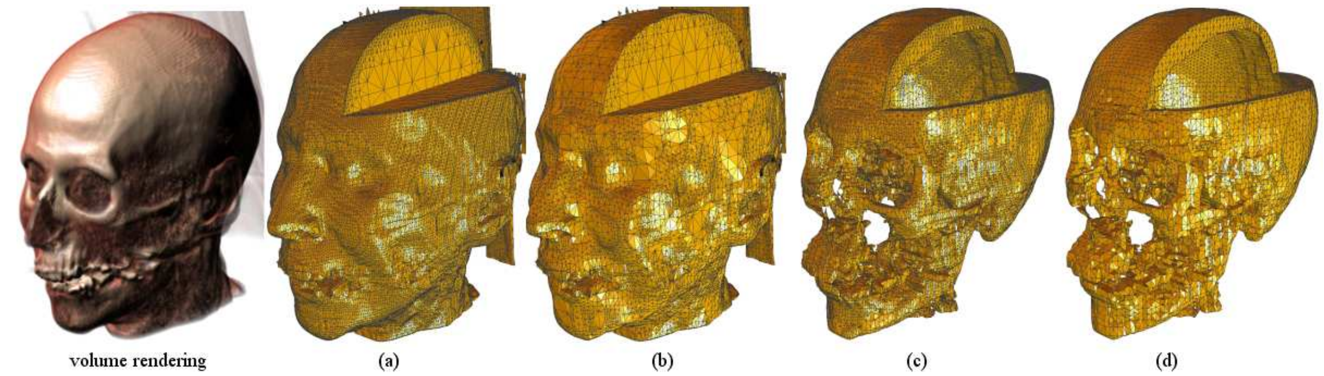

Figure 1: Adaptive tetrahedral meshes extracted from UNC Head (CT, 129×129×129). Isovalues (α_{in}, α_{out}) = (1000, 50) in (a)(b), and (1000, 120) in (c)(d); error tolerance ε_{in} = 0.0001, ε_{out} = (a): 0.0001, (b): 2.856, (c): 2.627, (d): 9.999. in **and** out **represent inner and outer isosurface respectively. The number of elements and the extraction time are listed in Figure 4.**

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 finite element simulations. Our comprehensive approach combines bilateral and anisotropic (feature specific) diffusion filtering, with contour spectrum based, isosurface and interval volume selection. Next, a top-down octree subdivision coupled with the dual contouring method is used to rapidly extract adaptive 3D finite element meshes from volumetric imaging data. The main contributions are extending the dual contouring method to crack free interval volume tetrahedralization and hexahedralization with feature sensitive adaptation. Compared to other tetrahedral extraction methods from imaging data, our method generates better quality adaptive 3D meshes without hanging nodes. Our method has the properties of crack prevention and feature sensitivity.

Categories and Subject Descriptors: I.3.5 [Computation Geometry and Object Modeling]: CSG - Curve, surface, solid and object representations

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

General Terms: Algorithms

Keywords: 3D meshes, adaptive, quality, feature sensitive, 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 (SDF) data. 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.

For accurate and efficient Finite Element Method (FEM) calculations, it is important to have accurate and high quality models, minimize the number of elements and preserve features. In this paper, we present a comprehensive approach 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, the main steps to extract tetrahedral/hexahedral meshes from the interval volume, I_F, between the two isosurfaces (Equation (1)) are as follows:

1. Volumetric Denoising
2. Contour spectrum based interval volume selection.

3. Adaptive 3D meshing with feature sensitivity.
4. Quality improvement

As a preprocessing step, the bilateral prefiltering coupled with anisotropic diffusion method [2] is applied to volumetric data. Accurate gradient estimation can also be obtained. The Contour Spectrum [1] 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 [11] 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) [9].

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

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. 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 byproduct, 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.

In Dual Contouring, QEF is used for isosurface extraction and sharp features can be preserved. But how to identify features such as sharp edges and facial features (like nose, eyes, mouth and ears)? This paper adopts a different error function to identify those features sensitively. 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 work; Section 3 introduces the outline of our comprehensive 3D mesh extracting method. Section 4 explains the detailed algorithm of how to extract tetrahedra and hexahedra from 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, and presents our conclusion.

2. PREVIOUS WORK

In the last twenty years, the techniques of CT and MRI have developed rapidly. Computer visualization, and engineering calculation (FEM) require certain kinds of mesh extracted from these scanned volume data.

Multiresolution Isosurface Extraction The predominant algorithm for isosurface extraction from volume data is Marching Cubes (MC) [14], which computes a local triangulation within each cube to approximate the isosurface by using a case table of edge intersections. Furthermore, the asymptotic decider was proposed to avoid ambiguities existing in MC [17] [13]. For efficient isosurface extraction, [3] starts from seed cells and traces the rest of the isosurface components by contour propagation.

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 [21]. A surface wave-front propagation technique [22] is used to generate multiresolution meshes with good aspect ratio. By combining SurfaceNets [10] and the extended Marching Cubes algorithm [12], octree based Dual Contouring [11] 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 [12] were introduced to extract feature sensitive isosurfaces from volume data. The grid snapping method reduces the number of elements in an approximated isocontour and also improves the aspect ratio of the elements [16]. [4] studied how to generate triangular meshes with bounded aspect ratios from a planar point set. [15] proposed an algorithm to triangulate a d-dimensional region with a bounded aspect ratio.

Quality Meshing MC is extended to extract tetrahedral meshes between two isosurfaces directly from volume data [8]. A different and systematic algorithm, Marching Tetrahedra (MT), was proposed for interval volume tetrahedralization [18]. A multiresolution 3D meshes [23] can be generated by combining recursive subdivision and edge-bisection methods. Poor quality tetrahedra called slivers are notoriously common in 3D Delaunay triangulations. Sliver exudation [7] is used to eliminate those slivers. A deterministic algorithm [6] was presented for generating a weighted Delaunay mesh with no poor quality tetrahedra including slivers. Shewchuk [20] provides some valuable conclusions on quality measures for FEM.

3. OVERVIEW

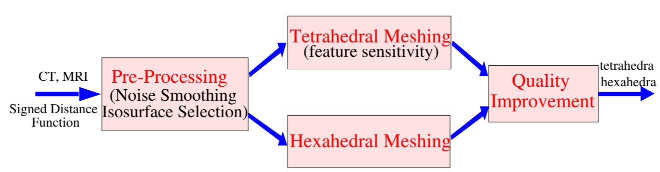

Figure 2: Overview for 3D mesh extraction

Our comprehensive 3D meshing method is displayed in Figure 2. 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 3D meshes from the interval volume, and a feature sensitive error function is adopted to reduce the number of elements while preserving features. Finally, the edge contraction method is used to improve the mesh quality.

Since noise influences the accuracy of the extracted meshes, it is important to remove it before the mesh extracting process. We use the anisotropic diffusion method [2] to smooth noise. In order to obtain more accurate computation of curvature and gradient for the 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.

Mesh extraction from imaging data requires selecting suitable boundary isosurfaces. We use a user interface called Contour Spectrum [1], to find isosurfaces of interest. The Contour Spectrum computes quantitative properties such as surface area, volume, and gradient integral of contours, and helps to choose suitable isosurfaces by showing the related spectrum in a 2D plane. A contour tree [5] can be used to capture the topological information on each isosurface and help choose isosurfaces with desirable topology.

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. 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. Here are definitions used in the algorithm description.

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.

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

Interior Cell Different from the boundary cell, all the eight vertices of an interior cell lie inside the interval volume.

Interior Face in Boundary Cell In the boundary cell, those faces with all four vertices lying inside the interval volume are called interior faces.

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 (e.g. T-Vertex).

4.1 Uniform Tetrahedral Extraction

For isosurface extraction, we only need to analyze boundary cells – those cells that contain sign change edges. There are four neighbor cubes which share the same sign change edge. Dual Contouring generates one minimal vertex for each neighbor cube by minimizing the QEF, then connects them to generate a quad. By marching all sign change edges, the isosurface is obtained. For tetrahedral mesh extraction, cells inside the interval volume should also be set as leaves besides boundary cells.

4.1.1 Uniform 2D Triangulation

Figure 3(1) 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 QEF minimizers of two cells which share the edge. The two minimizers and the interior vertex of the edge construct a triangle (blue).

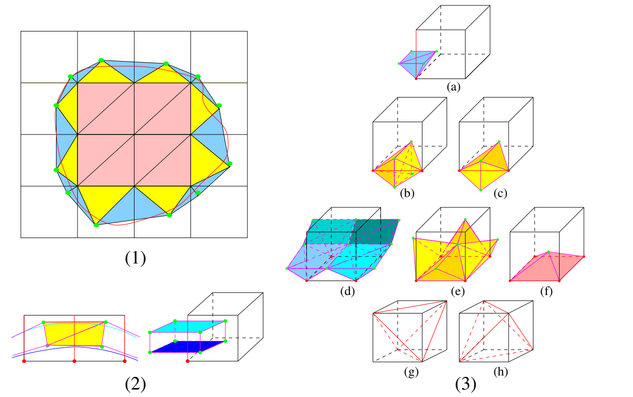

Figure 3: (1) - Uniform Triangulation, the red curve represents the isocontour. (2) - Sign Change Edge Passed Across by Two Isosurfaces, Left (2D) : the cyan and blue curves represent the two isocontours; Right (3D): the cyan and blue quads approximate the two isosurfaces. The red edges are sign change edges. (3) - Case Table of Uniform Tetrahedralization - the red vertex means it lies interior to the interval volume. In (1)(2)(3), green points represent minimizers.

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

4.1.2 Uniform 3D Tetrahedralization

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 3(3a), 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 3(3b)(3c), 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 (3c) assumes the cell below this boundary cell is interior to 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 3(3f)). Figure 3(3d)(3e)(3f) give a sequence how to generate tetrahedra when there is only one interior face in the boundary cell. (3d) analyzes four sign change edges, (3e) deals with four interior edges and (3f) fills the gap.
4. Interior cell – decompose the interior cube into five tetrahedra. There are two different decomposition ways (Figure 3(3g)(3h)). For two adjacent cells, we choose a different decomposition method to avoid the diagonal choosing conflict problem.

If two isosurfaces pass across the same sign change edge, we can split the cell into eight cubes in the octree data structure, then analyze each small cubes separately. In another approach, we need to analyze the sign change edge twice (Figure 3(2)) and fill gaps in the boundary cell. In 2D, two minimizers are obtained for the inner isosurface, and similarly two minimizers are calculated for the outer isosurface. They construct a quad, which can be decomposed into two triangles. For 3D, 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 vectors. This can be solved by splitting the cell into eight cubes.

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 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 method provides 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.

4.2.1 Adaptive 2D Triangulation

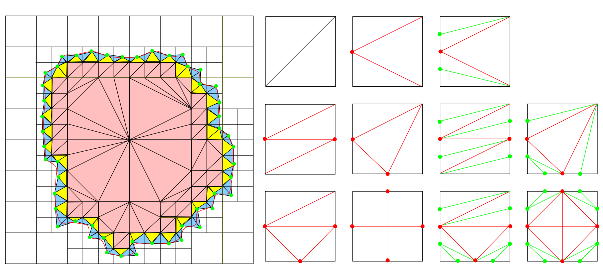

Figure 5: Left: Adaptive Triangulation. The red curve represents the isocontour, green points represent minimizers. Right: Case Table for Decomposing the Interior Cell into Triangles. Red points and red lines mean its neighbors have level (κ+1); green points and green lines mean its neighbors have a higher level than (κ+1).

Figure 5(left) gives an example of how to triangulate the interior area of an isocontour. Similarly, we need to analyze the following problems:

1. Sign change edge – if the edge is minimal, deal with it as in the uniform case (blue triangles).
2. Interior edge in boundary cell – if the edge is minimal, analyze it as in the uniform case (yellow triangles).
3. Interior cell – Figure 5 (right) lists all the cases of how to decompose the interior cell into triangles. In order to obtain triangles with good aspect ratio, we restrict the neighboring level difference to be ≤ 2.

Compared to the uniform case, the triangulation of interior cells is more complicated (Figure 5). 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 ith 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 a higher level than (κ+1). So we need to search all the middle points on this edge. A top-down or a 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. 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.

4.2.2 Adaptive 3D Tetrahedralization

For 3D adaptive tetrahedralization, we use the same algorithm with the uniform case to analyze the boundary cell.

1. Sign change edge – if the edge is minimal, deal with it as in the uniform case.
2. Interior edge in the boundary cell – if the edge is minimal, deal with it as in the uniform case.
3. Interior face in boundary cell – identify all middle points on the four edges, and decompose the face into triangles 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 5), then insert a Steiner point at the cell center. Each triangle and the Steiner point construct a tetrahedron.

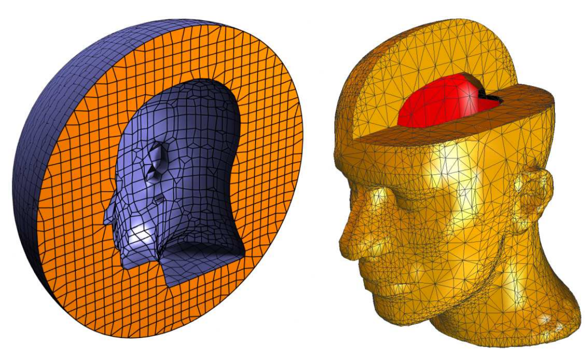

Figure 6: Left - hexahedralization of the volume between the human head and a sphere boundary; Right - an adaptive tetrahedral mesh.

Data Set	Type	Resolution	Number of Tetrahedra (Extraction Time (unit : ms))			
			(a)	(b)	(c)	(d)
UNC Head (Skin)	CT	$129 \times 129 \times 129$	935124 (17406)	545269 (10468)	–	–
UNC Head (Skull)	CT	$129 \times 129 \times 129$	–	–	579834 (10203)	166271 (3063)
Poly	CT	$257 \times 257 \times 257$	276388 (5640)	63325 (1672)	14204 (672)	–
Knee	SDF	$65 \times 65 \times 65$	70768 (1360)	94586 (1782)	93330 (1750)	72366 (1406)

Figure 4: Data Sets and Test Results. The CT data sets are re-sampled to fit into the octree representation (Figure 1, 9).

By using the above algorithm, we extract tetrahedral meshes from volumetric imaging data successfully. Figure 6 (right) gives one example.

4.3 Hexahedral Extraction

Finite element calculations sometimes require hexahedral 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 6 (left) shows an example used to solve electromagnetic scattering simulations.

5. ERROR METRIC

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 while preserving features, 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 some features, for example, facial features, like the nose, eyes, mouth and ears of the human head model in Figure 7. Here we choose the Euclidean distance error (EDerror) function to identify features.

For level (i), the eight vertices' function values are given, and a trilinear function is defined in Equation (4), from which the function values of 12 edge middle points, 6 face middle points and 1 center point can be obtained. For level (i+1), the function values of all vertices are given. The error function is defined in Equation (5).

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

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

The two error functions are compared in Figure 7. It is obvious that the EDerror can also preserve sharp edges, and is more sensitive to the areas where nose, eyes, mouth and ears are located on the human head model. That is because EDerror is the Euclean distance which measures

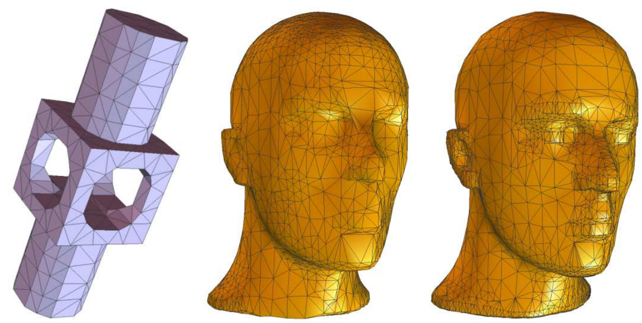

Figure 7: Sharp edge features (left); Facial features: QEF (middle, 2952 triangles) and EDerror (right, 2734 triangles). Better feature adaptation (eyes, nose, mouth and ears) is shown in the right picture.

error in a better way [19] than QEF. Furthermore, QEF only measures function value at a minimizer point for each cell, while EDerror compares function value at all vertices and edge/face middle points.

6. QUALITY IMPROVEMENT

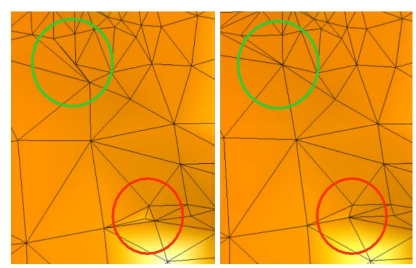

Figure 8: Quality Improvement - Left: no edge contraction, circles mark triangles with bad aspect ratio; Right: poor quality triangles disappear after iterative edge contraction.

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

- Tetrahedral Quality Measure = volume of tetrahedron / volume of equilateral tetrahedron with same circumsphere radius (> 0.02)
- Min/Max Angles – with minimum angle $\alpha > 10^\circ$ and maximum angle $\beta < 160^\circ$.
- Right-hand-side principle

In the process of improving the mesh quality, edge contraction is a direct method to eliminate sliver tetrahedra. For each tetrahedron, first calculate the three quality parameters. If the tetrahedron's orientation is Left-hand-side, swap any two vertices' index number. If Tetrahedral Quality Measure ≤ 0.02 or Min Angle $\leq 10°$ or Max Angle $\geq 160°$, contract the shortest edge. Be careful not to merge vertices on surfaces to vertices inside the interval volume. Figure 8 shows an example.

7. RESULTS AND CONCLUSION

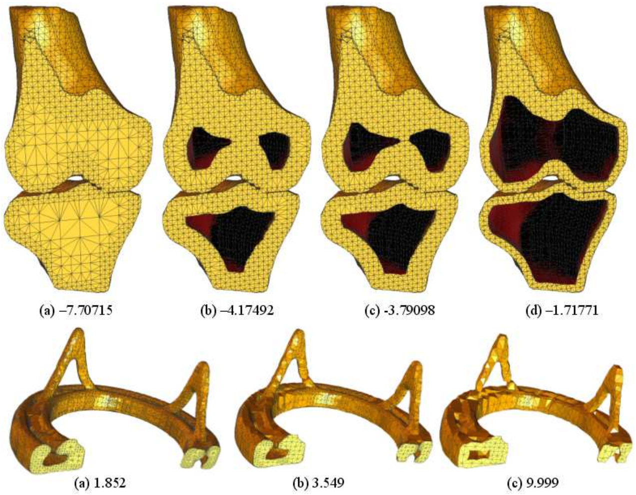

Figure 9: Upper row: Knee (SDF) – error tolerances ε_{in} = ε_{out} = 0.0001; isovalues α_{out} = -0.02838, α_{in} are listed below each picture. Bottom row: Heart Valve (Poly, CT) – isovalues (α_{in}, α_{out}) = (1000, 75); error tolerances ε_{in} = 0.0001, ε_{out} are listed below each picture.

We developed an interactive program for 3D mesh extraction and rendering from a volume. 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.

Figure 4, 9 provide information about data sets and test results. 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 SDF from the surface and performed mesh extraction (Figure 9 (knee)). The results from CT data are shown in Figure 1 and 9 (heart valve). The number of elements is controlled by changing error tolerance. In Figure 9 (upper row), the sequence of images are generated by changing the isovalue of the inner isosurface. The topology of the inner isosurface can change arbitrarily.

We have presented an algorithm to extract adaptive and high quality 3D meshes directly from volumetric imaging data. By extending the dual contouring method [11], our method can generate 3D meshes with good properties such as 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.

8. ACKNOWLEDGMENTS

We thank S. Khandelwal, V. Siddavanahalli, J. Sun, A. Thane, and Q. Wu for their help, 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.

9. REFERENCES

[1] C. Bajaj, V. Pascucci, and D. Schikore. The contour spectrum. In *Proceeding of IEEE Visualization 1997*, pages 167–174.

[2] C. Bajaj, Q. Wu, and G. Xu. Level set based volume anisotropic diffusion. In *ICES Technical Report 301, The Univ. of Texas at Austin, www.ticam.utexas.edu/ccv/papers/*, 2003.

[3] Chandrajit L. Bajaj, Valerio Pascucci, and Daniel Schikore. Fast isocontouring for improved interactivity. In *Proceedings of the IEEE Symposium on Volume Visualization, ACM Press*, pages 39–46, 1996.

[4] Marshall W. Bern, David Eppstein, and John R. Gilbert. Provably good mesh generation. In *IEEE Symposium on Foundations of Computer Science*, pages 231–241, 1990.

[5] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry : Theory and Applications*, 24(2):75–94, 2003.

[6] S.-W. Cheng and T. K. Dey. Quality meshing with weighted delaunay refinement. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms (SODA 2002)*, pages 137–146, 2002.

[7] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S. Teng. Sliver exudation. *Journal of ACM*, 47:883–904, 2000.

[8] I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima. Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):144–155, 1996.

[9] Michael Garland and Paul S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization '98*, pages 263–270, 1998.

[10] Sarah F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 23–30.

[11] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *ACM SIGGRAPH 2002*, pages 339 – 346.

[12] L. Kobbelt, M. Botsch, U. Schwanecke, and H. Seidel. Feature sensitive surface extraction from volume data. In *SIGGRAPH 2001*, pages 57–66.

[13] A. Lopes and K. Brodlie. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. In *IEEE Transactions on Visualization and Computer Graphics*, pages 19–26, 2003.

[14] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH 1987*, pages 163–169, 1987.

[15] Scott A. Mitchell and Stephen A. Vavasis. Quality mesh generation in higher dimensions. *SIAM Journal on Computing*, 29(4):1334–1370, 2000.

[16] D. Moore. Compact isocontours from sampled data. *Graphics Gems III*, pages 23–28, 1992.

[17] G. M. Nielson and B. Hamann. The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proceedings of Visualization '91*, pages 83–90, 1991.

[18] G. M. Nielson and J. Sung. Interval volume tetrahedrization. In *IEEE Visualization '97*, pages 221–228, 1997.

[19] Vaughan Pratt. Direct least-squares fitting of algebraic surfaces. In *ACM SIGGRAPH 1987*, pages 145–152.

[20] J. R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings of International Meshing Roundtable*, pages 115–126, 2002.

[21] R. Westermann, L. Kobbelt, and T. Ertl. Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer*, 15(2):100–111, 1999.

[22] Z. Wood, M. Desbrun, P. Schroder, and D.E. Breen. Semi-regular mesh extraction from volumes. In *Visualization 2000 Conference Proceedings*, pages 275–282, 2000.

[23] Yong Zhou, Baoquan Chen, and A. Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In *IEEE Visualization*, pages 135–142, 1997.