COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng 2000; **00**:1–6 *Prepared using* cnmauth.cls *[Version: 2002/09/18 v1.02]*

Surface Smoothing and Quality Improvement of Quadrilateral/Hexahedral Meshes with Geometric Flow[†]

Yongjie Zhang[†] Chandrajit Bajaj[†] Guoliang Xu[§]

[†] *Computational Visualization Center, Department of Computer Sciences and Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA*

[§] *State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China*

SUMMARY

This paper describes an approach to smooth the surface and improve the quality of quadrilateral/hexahedral meshes with feature preserved using geometric flow. For quadrilateral surface meshes, the surface diffusion flow is selected to remove noise by relocating vertices in the normal direction, and the aspect ratio is improved with feature preserved by adjusting vertex positions in the tangent direction. For hexahedral meshes, besides the surface vertex movement in the normal and tangent directions, interior vertices are relocated to improve the aspect ratio. Our method has the properties of noise removal, feature preservation and quality improvement of quadrilateral/hexahedral meshes, and it is especially suitable for biomolecular meshes because the surface diffusion flow preserves sphere accurately if the initial surface is close to a sphere. Several demonstration examples are provided from a wide variety of application domains. Some extracted meshes have been extensively used in finite element simulations. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: quadrilateral/hexahedral mesh, surface smoothing, feature preservation, quality improvement, geometric flow.

1. INTRODUCTION

The quality of unstructured quadrilateral/hexahedral meshes plays an important role in finite element simulations. Although a lot of efforts have been made, it still remains a challenging problem to generate quality quad/hex meshes for complicated structures such as the biomolecule Ribosome 30S shown in Figure 1. We have described an isosurface extraction method to generate quad/hex meshes for arbitrary complicated structures from volumetric data and utilized an optimization-based method to improve the mesh quality [1, 2], but the surface needs to be smoothed and the mesh quality needs to be further improved.

[†]Visit `http://ccvweb.csres.utexas.edu/ccv/meshing/quadhexgf`
Email addresses: jessica@ices.utexas.edu (Yongjie Zhang), bajaj@cs.utexas.edu (Chandrajit Bajaj), xuguo@lsec.cc.ac.cn (Guoliang Xu)

Received 17 January 2000
Revised 18 September 2002

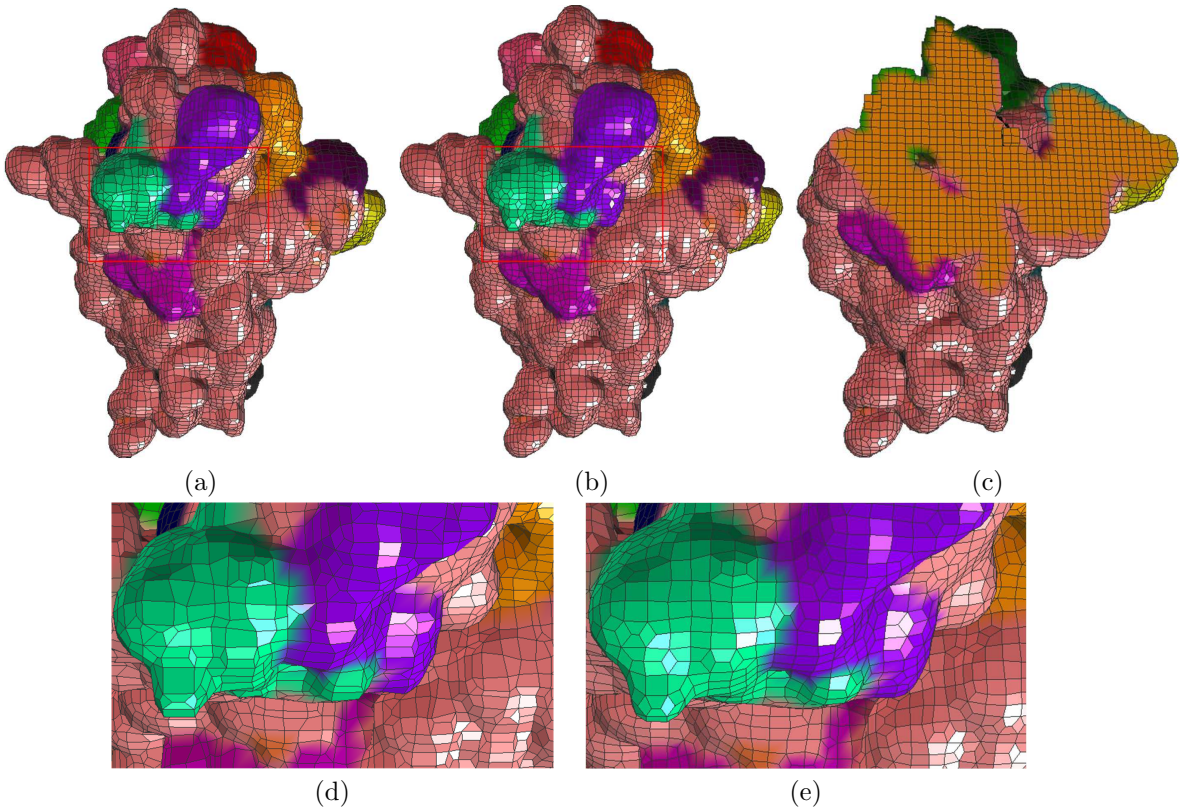


Figure 1. The comparison of mesh quality of Thermus Thermophilus small Ribosome 30S (1J5E) crystal subunit. The pink color shows 16S rRNA and the remaining colors are proteins. (a) the original quadrilateral mesh (13705 vertices, 13762 quads); (b) the improved quadrilateral mesh; (c) the improved hexahedral mesh (40294 vertices, 33313 hexes); (d) the zoom-in picture of the red box in (a); (e) the zoom-in picture of the red box in (b). The mesh quality is measured by three quality metrics as shown in Figure 10.

Geometric partial differential equations (GPDEs) such as Laplacian smoothing have been extensively used in surface smoothing and mesh quality improvement. There are two main methods in solving GPDEs, the finite element method (FEM) and the finite difference method (FDM). Although FDM is not robust sometimes, people still prefer to choosing FDM instead of FEM because FDM is simpler and easier to implement. Recently, a discretized format of the Laplacian-Beltrami (LB) operator over triangular meshes was derived and used in solving GPDEs [3] [4] [5]. In this paper, we will discretize the LB operator over quadrilateral meshes, and discuss an approach to apply the discretizated format on surface smoothing and quality improvement for quadrilateral or hexahedral meshes.

The main steps to smooth the surface and improve the quality of quadrilateral and hexahedral meshes are as follows:

1. Discretizing the LB operator and denoising the surface mesh - vertex adjustment in the normal direction with volume preservation.

2. Improving the aspect ratio of the surface mesh - vertex adjustment in the tangent direction with feature preservation.
3. Improving the aspect ratio of the volumetric mesh - vertex adjustment inside the volume.

For quadrilateral meshes, generally only Step 1 and Step 2 are required, but all the three steps are necessary for surface smoothing and quality improvement of hexahedral meshes.

Unavoidly the quadrilateral or hexahedral meshes may have some noise over the surface, therefore the surface mesh needs to be smoothed. In this paper, we derive a discretized format of the LB operator, and choose the surface diffusion flow (Equation (1)) to smooth the surface mesh by relocating vertices along their normal directions. The surface diffusion flow is volume preserving and also preserves a sphere accurately if the initial surface mesh is embedded and close to a sphere, therefore it is especially suitable for surface smoothing of biomolecular meshes since biomolecules are usually modelled as a union of hard spheres.

The aspect ratio of the surface mesh can be improved by adjusting vertices in the tangent plane, and surface features are preserved since the movement in the tangent plane doesn't change the surface shape ([6], page 72). For each vertex, the mass center is calculated to find its new position on the tangent plane. Since the vertex tangent movement is an area-weighted relaxation method, it is also suitable for adaptive quadrilateral meshes.

Besides the movement of surface vertices, interior vertices also need to be relocated in order to improve the aspect ratio of hexahedral meshes. The mass center is calculated as the new position for each interior vertex.

Although our relaxation-based method can not guarantee that no inverted element is introduced for arbitrary input meshes, it works well in most cases with the properties of noise removal, feature preservation, mesh quality improvement. Furthermore, it is especially suitable for surface smoothing and quality improvement of biomolecular meshes. As the 'smart' Laplacian smoothing [7] [8], this method is applied only when the mesh quality is improved in order to avoid inverted elements. This method can also be combined with the optimization-based method to obtain a high quality mesh with relatively less computational cost.

The remainder of this paper is organized as follows: Section 2 reviews the previous related work; Sections 3 discusses the detailed algorithm of the LB operator discretization, surface smoothing and quality improvement of quadrilateral meshes; Sections 4 explains the quality improvement of hexahedral meshes; Section 5 shows some results and applications; The final section presents our conclusion.

2. PREVIOUS WORK

It is well-known that poor quality meshes result in poorly conditioned stiffness matrices in finite element analysis, and affect the stability, convergence, and accuracy of finite element solvers. Therefore, quality improvement is an important step in mesh generation.

Some quality improvement techniques of triangular and tetrahedral meshes, such as the edge-contraction method, can not be used for quadrilateral and hexahedral meshes because we do not want to introduce any degenerated elements. Therefore, the mesh smoothing methods are selected to improve the quality of quad/hex meshes by adjusting the vertex positions in the mesh while preserving its connectivity. As reviewed in [9] [10], Laplacian smoothing and optimization are the two main quality improvement techniques.

As the simplest and most straight forward method for node-based mesh smoothing, Laplacian smoothing relocates the vertex position at the average of the nodes connecting to it [11]. There are a variety of smoothing techniques based on a weighted average of the surrounding nodes and elements [12] [13] [14]. The averaging method may invert or degrade the local quality, but it is computationally inexpensive and very easy to implement, so it is in wide use. Winslow smoothing is more resistant to mesh folding because it requires the logical variables are harmonic functions [15].

Instead of relocating vertices based on a heuristic algorithm, people utilized an optimization technique to improve mesh quality. The optimization algorithm measures the quality of the surrounding elements to a node and attempts to optimize it [16]. The algorithm is similar to a minimax technique used to solve circuit design problems [17]. Optimization-based smoothing yields better results but it is more expensive than Laplacian smoothing, and it is difficult to decide the optimized iteration step length. Therefore, a combined Laplacian/optimization-based approach [7] [8] [18] was recommended. Physically-based simulations are used to reposition nodes [19]. Anisotropic meshes are obtained from bubble equilibrium [20] [21].

When we use the smoothing method to improve the mesh quality, it is also important to preserve surface features. Baker [22] presented a feature extraction scheme which is based on estimates of the local normals and principal curvatures at each mesh node. Local parametrization was utilized to improve the surface mesh quality while preserving surface characteristics [23], and two techniques called trapezium drawing and curvature-based mesh improvement were discussed in [24].

Staten et al. [25] [26] proposed algorithms to improve node valence for quadrilateral meshes. One special case of cleanup in hexahedral meshes for the whisker weaving algorithm is presented in [27]. Schneiders [28] proposed algorithms and a series of templates for quad/hex element decomposition. A recursive subdivision algorithm was proposed for the refinement of hex meshes [29].

3. QUADRILATERAL MESH

Noise may exist in quadrilateral meshes, therefore we need to smooth the surface mesh. The quality of some quadrilateral meshes may not be good enough for finite element calculations, and the aspect ratio also needs to be improved.

There are two steps for the surface smoothing and the quality improvement of quadrilateral meshes: (1) the discretization of Laplace-Beltrami opertor and the vertex movement along its normal direction to remove noise, (2) the vertex movement on its tangent plane to improve the aspect ratio while preserving surface features.

3.1. Geometric Flow

Various geometric partial differential equations (GPDEs), such as the mean curvature flow, the surface diffusion flow and Willmore flow, have been extensively used in surface and imaging processing [4]. Here we choose the surface diffusion flow to smooth the surface mesh,

$$\frac{\partial x}{\partial t} = \Delta H(x)\vec{n}(x). \tag{1}$$

where Δ is the Laplace-Beltrami (LB) operator, H is the mean curvature and $\vec{n}(x)$ is the unit normal vector at the node x. In [30], the existence and uniqueness of solutions for this flow was discussed, and the solution converges exponentially fast to a sphere if the initial surface is embedded and close to a sphere. It was also proved that this flow is area shrinking and volume preserving [4].

In applying geometric flows on surface smoothing and quality improvement over quadrilateral meshes, it is important to derive a discretized format of the LB operator. Discretized schemes of the LB operator over triangular meshes have been derived and utilized in solving GPDEs [3] [4] [5].

A quad can be subdivided into triangles, hence the discretization schemes of the LB operator over triangular meshes could be easily used for quadrilateral meshes. However, since the subdivision of each quad into triangles is not unique (there are two ways), the resulting discretization scheme is therefore not unique. Additionally in the discretization scheme, the element area needs to be calculated. If we choose to split each quad into two triangles and calculate the area of a quad as the summation of the area of two triangles, then the area calculated from the two different subdivisions could be very different because four vertices of a quad may not be coplanar. Therefore, a unique discretized format of the LB operator directly over quad meshes is required.

3.2. Discretized Laplace-Beltrami Operator

Here we will derive a discretized format for the LB operator over quadrilateral meshes. The basic idea of our scheme is to use the bilinear interpolation to derive the discretized format and to calculate the area of a quad. The discretization scheme is thus uniquely defined.

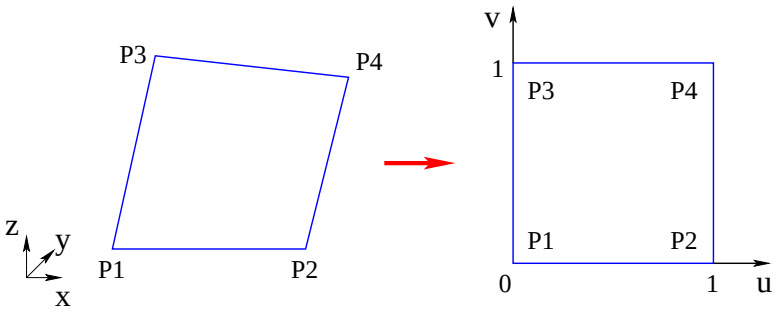


Figure 2. A quad $[p_1p_2p_4p_3]$ is mapped into a bilinear parametric surface.

Area Calculation: Let $[p_1p_2p_4p_3]$ be a quad in $\mathbb{R}^3$, then we can define a bilinear parametric surface S that interpolates four vertices of the quad as shown in Figure 2:

$$\begin{aligned} S(u,v) &= (1-u)(1-v)p_1 + u(1-v)p_2 \\ &+ (1-u)vp_3 + uvp_4. \end{aligned} \tag{2}$$

The tangents of the surface are

$$S_u(u,v) = (1-v)(p_2-p_1) + v(p_4-p_3), \tag{3}$$

$$S_v(u,v) = (1-u)(p_3-p_1) + u(p_4-p_2). \tag{4}$$

Let ∇ denote the gradient operator about the (x, y, z) coordinates of the vertex P_1, then we have

$$\begin{aligned} \nabla S_u(u,v) &= -(1-v), \quad (5) \\ \nabla S_v(u,v) &= -(1-u). \quad (6) \end{aligned}$$

Let A denote the area of the surface $S(u, v)$ for $(u, v) \in [0, 1]^2$, then we have

$$\begin{aligned} A &= \int_0^1 \int_0^1 \sqrt{\| S_u \times S_v \|^2} du dv \\ &= \int_0^1 \int_0^1 \sqrt{\| S_u \|^2 \| S_v \|^2 - (S_u, S_v)^2} du dv. \quad (7) \end{aligned}$$

It may not be easy to obtain the explicit form for integrals in calculating the area, numerical integration quadrature could be used. Here we use the following four-point Gaussian quadrature rule to compute the integral

$$\int_0^1 \int_0^1 f(u,v) du dv \approx \frac{f(q_1) + f(q_2) + f(q_3) + f(q_4)}{4}, \quad (8)$$

where

$$q^- = \frac{1}{2} - \frac{\sqrt{3}}{6}, \quad q^+ = \frac{1}{2} + \frac{\sqrt{3}}{6},$$

$$\begin{aligned} q_1 &= (q^-, \ q^-), \quad & q_2 &= (q^+, \ q^-), \\ q_3 &= (q^-, \ q^+), \quad & q_4 &= (q^+, \ q^+). \end{aligned}$$

The integration rule in Equation (8) is of $O(h^4)$, where h is the radius of the circumscribing circle.

Discretized LB Operator: The derivation of the discretized format of the LB operator is based on a formula in differential geometry [3]:

$$\lim_{diam(R) \to 0} \frac{2 \nabla A}{A} = \vec{H}(p), \quad (9)$$

where A is the area of a region R over the surface around the surface point p, $diam(R)$ denotes the diameter of the region R, and $\vec{H}(p)$ is the mean curvature normal.

From Equation (7), we have

$$\begin{aligned} \nabla A &= \int_0^1 \int_0^1 \nabla \sqrt{\| S_u \|^2 \| S_v \|^2 - (S_u, S_v)^2} du dv \\ &= \int_0^1 \int_0^1 \frac{S_u(S_v, (v-1)S_v - (u-1)S_u))}{\sqrt{\| S_u \|^2 \| S_v \|^2 - (S_u, S_v)^2}} du dv \\ &+ \int_0^1 \int_0^1 \frac{S_v(S_u, (u-1)S_u - (v-1)S_v)}{\sqrt{\| S_u \|^2 \| S_v \|^2 - (S_u, S_v)^2}} du dv \\ &= \alpha_{21}(p_2 - p_1) + \alpha_{43}(p_4 - p_3) \\ &+ \alpha_{31}(p_3 - p_1) + \alpha_{42}(p_4 - p_2), \quad (10) \end{aligned}$$

where

$$\begin{aligned}
\alpha_{21} &= \int_0^1\int_0^1 \frac{(1-v)(S_v,(v-1)S_v-(u-1)S_u))}{\sqrt{\| S_u \|^2 \| S_v \|^2 -(S_u,S_v)^2}}dudv \\
\alpha_{43} &= \int_0^1\int_0^1 \frac{v(S_v,(v-1)S_v-(u-1)S_u))}{\sqrt{\| S_u \|^2 \| S_v \|^2 -(S_u,S_v)^2}}dudv \\
\alpha_{31} &= \int_0^1\int_0^1 \frac{(1-u)(S_u,(u-1)S_u-(v-1)S_v)}{\sqrt{\| S_u \|^2 \| S_v \|^2 -(S_u,S_v)^2}}dudv \\
\alpha_{42} &= \int_0^1\int_0^1 \frac{u(S_u,(u-1)S_u-(v-1)S_v)}{\sqrt{\| S_u \|^2 \| S_v \|^2 -(S_u,S_v)^2}}dudv
\end{aligned}$$

∇A could be written as

$$\nabla A = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \alpha_4 p_4 \tag{11}$$

with

$$\begin{aligned}
\alpha_1 &= -\alpha_{21} - \alpha_{31}, \quad & \alpha_2 &= -\alpha_{21} + \alpha_{42}, \\
\alpha_3 &= \alpha_{31} - \alpha_{43}, \quad & \alpha_4 &= \alpha_{43} + \alpha_{42}.
\end{aligned} \tag{12}$$

Here we still use the four-point Gaussian quadrature rule in Equation (8) to compute the integrals in the α_{ij}. It follows from Equation (12) that $\sum_{i=1}^{4} \alpha_i = 0$, we have

$$\nabla A = \alpha_2(p_2 - p_1) + \alpha_3(p_3 - p_1) + \alpha_4(p_4 - p_1). \tag{13}$$

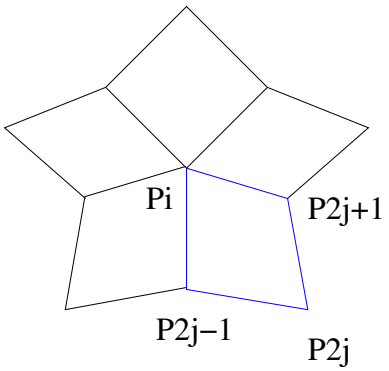


Figure 3. A neighboring quad $[p_i p_{2j-1} p_{2j} p_{2j+1}]$ around the vertex p_i.

Now let p_i be a vertex with valence n, and p_{2j} $(1 \le j \le n)$ be one of its neighbors on the quadrilateral mesh, then we can define three coefficients α_2, α_3, α_4 as in (13). Now we denote these coefficients as α_j^i, β_j^i and γ_j^i for the quad $[p_i p_{2j-1} p_{2j} p_{2j+1}]$ as shown in Figure 3. By

using Equation (13), the discrete mean curvature normal can be defined as

$$\begin{aligned}\vec{H}(p_i) &\approx \frac{2}{A(p_i)}\sum_{j=1}^{n}[\alpha_j^i(p_{2j-1}-p_i)\\ &+ \beta_j^i(p_{2j+1}-p_i)+\gamma_{j+1}^i(p_{2j}-p_i)] \\ &= \sum_{k=1}^{2n} w_k^i(p_k-p_i)\end{aligned} \tag{14}$$

where $\vec{H}(p_i)$ denotes the mean curvature normal, $A(p_i)$ is the total area of the quads around p_i, and

$$w_{2j}^i=\frac{2\gamma_j^i}{A(p_i)},\quad w_{2j-1}^i=\frac{2(\alpha_j^i+\beta_{j-1}^i)}{A(p_i)},\quad w_{2j+1}^i=\frac{2(\alpha_{j+1}^i+\beta_j^i)}{A(p_i)}.$$

Using the relation $\Delta x = 2H(p_i)$ ([31], page 151), we obtain

$$\Delta f(p_i) \approx 2\sum_{k=1}^{2n} w_k^i(f(p_k)-f(p_i)). \tag{15}$$

Therefore,

$$\begin{aligned}\Delta H(p_i)\vec{n}(p_i) &\approx 2\sum_{k=1}^{2n} w_k^i(H(p_k)-H(p_i))\vec{n}(p_i)\\ &= 2\sum_{k=1}^{2n} w_k^i\left[\vec{n}(p_i)\vec{n}(p_k)^T\vec{H}(p_k)-\vec{H}(p_i)\right],\end{aligned} \tag{16}$$

where $\vec{H}(p_k)$ and $\vec{H}(p_i)$ are further discretized by (14). Note that $\vec{n}(p_i)\vec{n}(p_k)^T$ is a 3×3 matrix.

Figure 4 shows one example of the molecule consisting of three amino acids (ASN, THR and TYR) with 49 atoms. The molecular surface was bumpy as shown in Figure 4(a) since there are some noise existing in the input volumetric data, the surface becomes smooth after the vertex normal movement as shown in Figure 4(b).

3.3. Tangent Movement

In order to improve the aspect ratio of the surface mesh, we need to add a tangent movement in Equation (1), hence the flow becomes

$$\frac{\partial x}{\partial t}=\Delta H(x)\vec{n}(x)+v(x)\vec{T}(x), \tag{17}$$

where $v(x)$ is the velocity in the tangent direction $\vec{T}(x)$. First we calculate the mass center $m(x)$ for each vertex on the surface, then project the vector $m(x)-x$ onto the tangent plane. $v(x)\vec{T}(x)$ can be approximated by $[m(x)-x]-\vec{n}(x)^T[m(x)-x]\vec{n}(x)$ as shown in Figure 5.

Mass Center: A mass center p of a region S is defined by finding $p\in S$, such that

$$\int_S \parallel y-p\parallel^2 d\sigma = min. \tag{18}$$

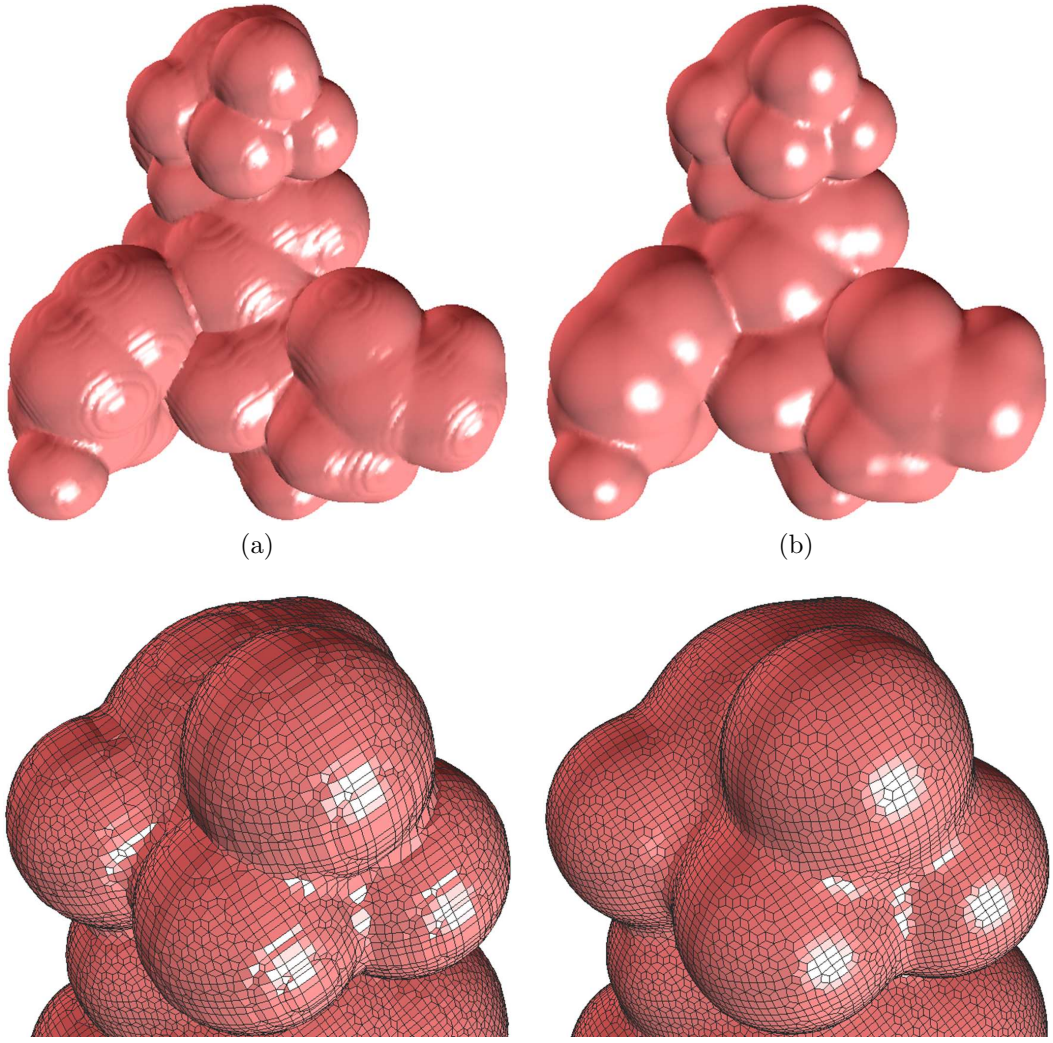


Figure 4. Surface smoothing and quality improvement of the molecule consisting of three amino acids (ASN, THR and TYR) with 49 atoms (45534 vertices, 45538 quads). (a) and (c) - the original mesh; (b) and (d) - after surface smoothing and quality improvement.

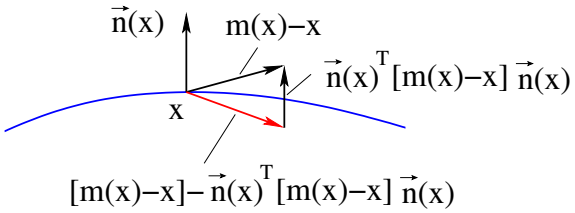


Figure 5. The tangent movement at the vertex x over a surface. The blue curve represents a surface, and the red arrow is the resulting tangent movement vector.

S is a piece of surface in $\mathbb{R}^3$, and S consists of quads around vertex x. Then we have

$$\sum(\frac{p_i + p_{2j-1} + p_{2j} + p_{2j+1}}{4} - p_i)A_j = 0, \tag{19}$$

A_j is the area of the quad $[p_i p_{2j-1} p_{2j} p_{2j+1}]$ calculated from Equation (7) using the integration rule in Equation (8). Then we can obtain

$$m(p_i) = \sum_{j=1}^{n}(\frac{p_i + p_{2j-1} + p_{2j} + p_{2j+1}}{4}A_j)/A^i_{total}, \tag{20}$$

where A^i_{total} is the total of quad areas around p_i. The area of a quad can be calculated using Equation (7).

In Figure 4, the vertex tangent movement is used to improve the aspect ratio of the quadrilateral mesh of the molecule consisting of three amino acids. Compared with Figure 4(c), it is obvious that the quadrilateral mesh becomes more regular and the aspect ratio is better as shown in Figure 4(d).

3.4. Temporal Discretization

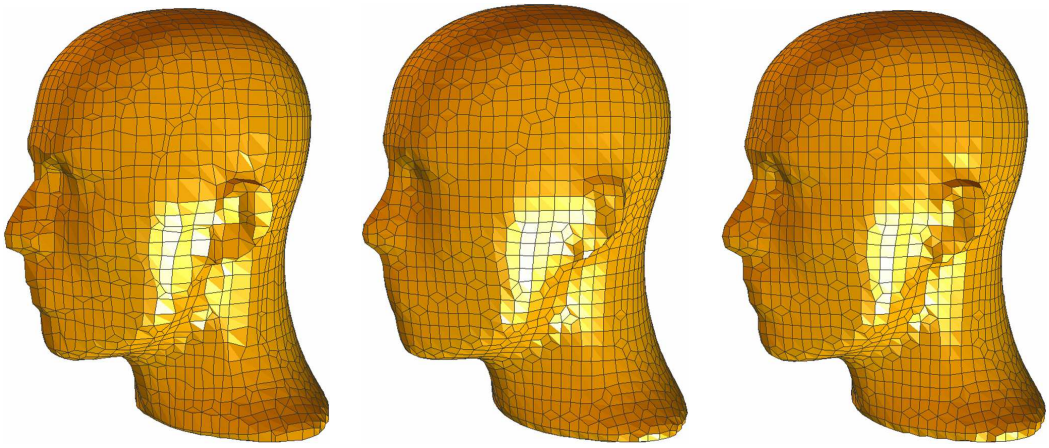
In the temporal space, $\frac{\partial x}{\partial t}$ is approximated by a semi-implicit Euler scheme $\frac{x_i^{n+1}-x_i^n}{\tau}$, where τ is the time step length. x_i^n is the approximating solution at $t = n\tau$, x_i^{n+1} is the approximating solution at $t = (n+1)\tau$, and x_i^0 serves as the initial value at x_i.

The spatial and temporal discretization leads to a linear system, and an approximating solution is obtained by solving it using a conjugate gradient iterative method with diagonal preconditioning.

3.5. Discussion

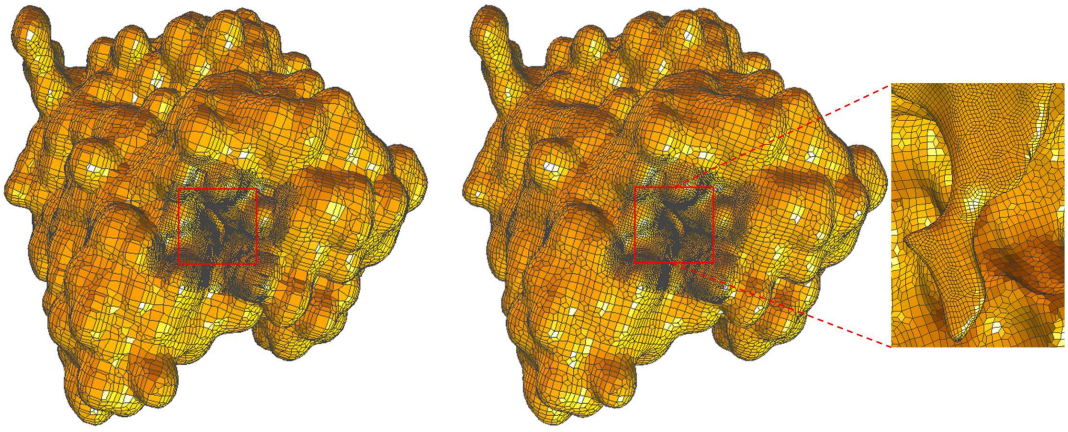
Vertex Normal Movement: The surface diffusion flow can preserve volume. Furthermore, it also preserves a sphere accurately if the initial mesh in embedded and close to a sphere. Suppose a molecular surface could be modelled by a union of hard spheres, so it is desirable to use the surface diffusion flow to evolve the molecular surface. Figure 4 shows one example, the molecular surface becomes more smooth and features are preserved after surface denoising.

Vertex Tangent Movement: If the surface mesh has no noise, we can only apply the tangent movement $\frac{\partial x}{\partial t} = v(x)\vec{T}(x)$ to improve the aspect ratio of the mesh while ignoring the vertex normal movement. Our tangent movement has two properties:



(a) (b) (c)

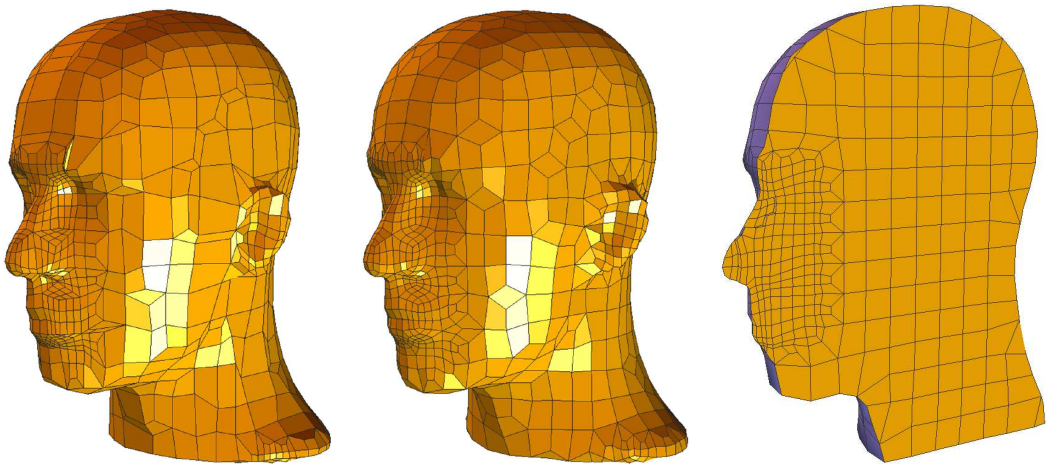
Figure 6. The quality of a quadrilateral mesh of a human head model is improved (2912 vertices, 2912 quads) after 100 iterations with the time step length 0.01. (a) The original mesh; (b) Each vertex is relocated to its mass center, some facial features are removed; (c) Only tangent movement is applied.



(a) (b)

Figure 7. The quality of an adaptive quadrilateral mesh of a biomolecule mAChE is improved (26720 vertices, 26752 quads). (a) the original mesh; (b) after quality improvement.

- The tangent movement doesn't change the surface shape ([6], page 72). Figure 6 shows the comparison of the human head model before and after the quality improvement. In Figure 6(b), each vertex is relocated to its mass center, so both normal movement and tangent movement are applied. After some iterations, the facial features, such as the nose, eyes, mouth and ears, are removed. In Figure 6(c), the vertex movement is restricted on the tangent plane, therefore facial features are preserved.
- The tangent movement is an area-weighted averaging method, which is also suitable for adaptive quad meshes as shown in Figure 7 and 8. In Figure 7, there is a cavity in the



(a) (b) (c)

Figure 8. Adaptive quadrilateral/hexadedral meshes of the human head. (a) the original quad mesh (1828 vertices, 1826 quads); (b) the improved quad mesh; (c) the improved hex mesh (4129 vertices, 3201 hexes), the right part of elements are removed to shown one cross section.

structure of biomolecule mouse acetylcholinesterase (mAChE), and denser meshes are generated around the cavity while coarser meshes are kept in all other regions. In Figure 8, finer meshes are generated in the region of facial features of the human head.

From Figure 6, 7 and 8, we can observe that after tangent movement, the quadrilateral meshes become more regular and the aspect ratio of the meshes is improved, as well as surface features are preserved.

4. HEXAHEDRAL MESH

There are three steps for surface smoothing and quality improvement of hexahedral meshes, (1) surface vertex normal movement, (2) surface vertex tangent movement and (3) interior vertex relocation.

4.1. Boundary Vertex Movement

The dual contouring hexahedral meshing method [2] [1] provides a boundary sign for each vertex and each face of a hexahedron, indicating if it lies on the boundary surface or not. For example, a vertex or a face is on the surface if its boundary sign is 1, while lies inside the volume if its boundary sign is 0.

The boundary sign for each vertex/face can also be decided by checking the connectivity information of the input hexahedral mesh. If a face is shared by two elements, then this face is not on the boundary; if a face belongs to only one hex, then this face lies on the boundary surface, whose four vertices are also on the boundary surface.

We can use the boundary sign to find the neighboring vertices/faces for a given vertex.

For each boundary vertex, we first find all its neighboring vertices and faces lying on the boundary surface by using the boundary sign, then relocate it to its new position calculated from Equation (17). There is a special situation that we need to be careful, a face/edge, whose four/two vertices are on the boundary, may not be a boundary face.

4.2. Interior Vertex Movement

For each interior vertex, we intend to relocate it to the mass center of all its surrounding hexahedra. There are different methods to calculate the volume for a hexahedron. Some people divide a hex into five or six tetrahedra, then the volume of the hex is the summation of the volume of these five or six tetrahedra. This method is not unique since there are various dividing formats. Here we use an trilinear parametric function to calculate the volume of a hex.

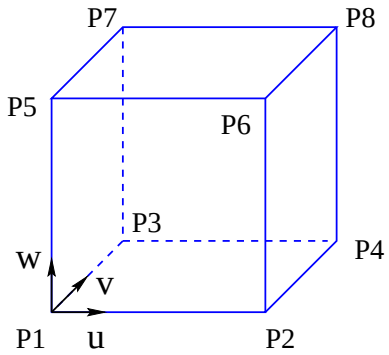


Figure 9. The trilinear parametric volume V of a hexahedron $[p_1p_2\dots p_8]$.

Volume Calculation: Let $[p_1p_2\dots p_8]$ be a hex in $\mathbb{R}^3$, then we define the trilinear parametric volume $V(u,v,w)$ that interpolates eight vertices of the hex as shown in Figure 9:

$$\begin{aligned} V(u,v,w) &= (1-u)(1-v)(1-w)p_1 \\ &+ u(1-v)(1-w)p_2 + (1-u)v(1-w)p_3 \\ &+ uv(1-w)p_4 + (1-u)(1-v)wp_5 \\ &+ u(1-v)wp_6 + (1-u)vwp_7 \\ &+ uvwp_8. \end{aligned} \tag{21}$$

The tangents of the volume are

$$\begin{aligned} V_u(u,v,w) &= (1-v)(1-w)(p_2-p_1) + v(1-w)(p_4-p_3) \\ &+ (1-v)w(p_6-p_5) + vw(p_8-p_7), \\ V_v(u,v,w) &= (1-u)(1-w)(p_3-p_1) + u(1-w)(p_4-p_2) \\ &+ (1-u)w(p_7-p_5) + uw(p_8-p_6), \\ V_w(u,v,w) &= (1-u)(1-v)(p_5-p_1) + u(1-v)(p_6-p_2) \\ &+ (1-u)v(p_7-p_3) + uv(p_8-p_4). \end{aligned}$$

Let V denote the volume of $V(u,v,w)$ for $(u,v,w) \in [0,1]^3$, then we have

$$V = \int_0^1 \int_0^1 \int_0^1 \sqrt{\bar{V}} du dv dw \tag{22}$$

where

$$\bar{V} \quad = \quad \| \, (V_u \times V_v) \cdot V_w \, \|^2 \tag{23}$$

Numerical integration quadrature could be used. Here we choose the following eight-point Gaussian quadrature rule to compute the integral

$$\int_0^1 \int_0^1 \int_0^1 f(u,v,w) du dv dw \approx \frac{\sum_{j=1}^{8} f(q_j)}{8}, \tag{24}$$

where

$$q^- = \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad q^+ = \frac{1}{2} + \frac{\sqrt{3}}{6},$$

$$\begin{aligned} q_1 &= (q^-, \ q^-, \ q^-), & q_2 &= (q^+, \ q^-, \ q^-), \\ q_3 &= (q^-, \ q^+, \ q^-), & q_4 &= (q^+, \ q^+, \ q^-), \\ q_5 &= (q^-, \ q^-, \ q^+), & q_6 &= (q^+, \ q^-, \ q^+), \\ q_7 &= (q^-, \ q^+, \ q^+), & q_8 &= (q^+, \ q^+, \ q^+). \end{aligned}$$

The integration rule in Equation (24) is of $O(h^4)$, where h is the radius of the circumscribing sphere.

Mass Center: A mass center p of a region V is defined by finding $p \in V$, such that

$$\int_V \| \, y - p \, \|^2 \, d\sigma = min. \tag{25}$$

V is a piece of volume in $\mathbb{R}^3$, and V consists of hexahedra around vertex x. Then we have

$$\sum (\frac{1}{8} \sum_{j=1}^{8} p_j - p_i) V_j = 0, \tag{26}$$

V_j is the volume of the hex $[p_1 p_2 \dots p_8]$ calculated from the trilinear function, then we can obtain

$$m(p_i) = \sum_{j \in N(i)} (\frac{1}{8} \sum_{j=1}^{8} p_j V_j) / V_{total}^i, \tag{27}$$

where $N(i)$ is the index set of the one ring neighbors of p_i, and V_{total}^i is the total of hex volume around p_i.

The same Euler scheme is used here for temporal discretization, and the linear system is solved using the conjugate gradient iterative method.

Type	DataSet	MeshSize (Vertex♯, Elem♯)	Scaled Jacobian (best,aver.,worst)	Condition Number (best,aver.,worst)	Oddy Metric (best,aver.,worst)	Inverted Elem♯
quad	Head^1	(2912, 2912)	(1.0, 0.92, 0.02)	(1.0, 1.13, 64.40)	(0.0, 1.74, 8345.37)	0
	Head^2	-	(1.0, 0.93, 0.16)	(1.0, 1.11, 6.33)	(0.0, 0.63, 78.22)	0
	Head^3	-	(1.0, 0.96, 0.47)	(1.0, 1.05, 2.12)	(0.0, 0.22, 6.96)	0
	Ribosome 30S^1	(13705, 13762)	(1.0, 0.90, 0.03)	(1.0, 1.17, 36.90)	(0.0, 1.38, 2721.19)	0
	Ribosome 30S^2	-	(1.0, 0.90, 0.03)	(1.0, 1.17, 34.60)	(0.0, 1.37, 2392.51)	0
	Ribosome 30S^3	-	(1.0, 0.93, 0.06)	(1.0, 1.08, 16.14)	(0.0, 0.38, 519.22)	0
	Vhmale^1	(76975, 77708)	(1.0, 0.92, 0.01)	(1.0, 1.15, 97.91)	(0.0, 2.49, 19172.46)	0
	Vhmale^2	-	(1.0, 0.95, 0.02)	(1.0, 1.07, 48.29)	(0.0, 0.71, 4662.72)	0
	Vhmale^3	-	(1.0, 0.95, 0.04)	(1.0, 1.05, 24.83)	(0.0, 0.54, 1230.85)	0
hex	Head^1	(8128, 6587)	(1.0, 0.91, 1.7e-4)	(1.0, 2.99, 6077.33)	(0.0, 29.52, 1.80e5)	2
	Head^2	-	(1.0, 0.91, 0.005)	(1.0, 1.96, 193.49)	(0.0, 6.34, 5852.23)	0
	Head^3	-	(1.0, 0.92, 0.007)	(1.0, 1.80, 147.80)	(0.0, 4.50, 1481.69)	0
	Ribosome 30S^1	(40292, 33313)	(1.0, 0.91, 2.4e-5)	(1.0, 2.63, 4.26e4)	(0.0, 34.15, 2.27e6)	5
	Ribosome 30S^2	-	(1.0, 0.91, 0.004)	(1.0, 1.74, 263.91)	(0.0, 4.97, 8017.39)	0
	Ribosome 30S^3	-	(1.0, 0.92, 0.004)	(1.0, 1.59, 237.36)	(0.0, 3.42, 5133.25)	0

Figure 10. The comparison of the three quality criteria (the scaled Jacobian, the condition number and Oddy metric) before/after the quality improvement for quad/hex meshes of the human head (Figure 12) and Ribosome 30S (Figure 1). DATA^1 – before quality improvement; DATA^2 – after quality improvement using the optimization scheme in [1]; DATA^3 – after quality improvement using the combined geometric flow/optimization-based approach.

5. RESULTS AND APPLICATIONS

There are many different ways to define the aspect ratio for a quad or a hex to measure the mesh quality. Here we choose the scaled Jacobian, the condition number of the Jacobian matrix and Oddy metric [32] as our metrics [33][34][35].

Assume $x \in \mathbb{R}^3$ is the position vector of a vertex in a quad or a hex, and $x_i \in \mathbb{R}^3$ for $i = 1, \ldots, m$ are its neighboring vertices, where $m = 2$ for a quad and $m = 3$ for a hex. Edge vectors are defined as $e_i = x_i - x$ with $i = 1, \ldots, m$, and the Jacobian matrix is $J = [e_1, ..., e_m]$. The determinant of the Jacobian matrix is called *Jacobian*, or *scaled Jacobian* if edge vectors are normalized. An element is said to be *inverted* if one of its Jacobians ≤ 0. We use the *Frobenius norm* as a matrix norm, $|J| = (tr(J^T J)^{1/2})$. The condition number of the Jacobian matrix is defined as $\kappa(J) = |J||J^{-1}|$, where $|J^{-1}| = \frac{|J|}{det(J)}$. Therefore, the three quality metrics for a vertex x in a quad or a hex are defined as follows:

$$Jacobian(x) = det(J) \tag{28}$$

$$\kappa(x) = \frac{1}{m}|J^{-1}||J| \tag{29}$$

$$Oddy(x) = \frac{(|J^T J|^2 - \frac{1}{m}|J|^4)}{det(J)^{\frac{4}{m}}} \tag{30}$$

where $m = 2$ for quadrilateral meshes and $m = 3$ for hexahedral meshes.

In [1], an optimization approach was used to improve the quality of quad/hex meshes. The goal is to remove all the inverted elements and improve the worst condition number of the Jacobian matrix. Here we combine our surface smoothing and quality improvement schemes with the optimization-based approach. We use the geometric flow to improve the quality of quad/hex meshes overall and only use the optimization-based smoothing when necessary. Figure 10 shows the comparison of the three quality criteria before and after quality improvement. We can observe that the aspect ratio is improved using the combined approach.

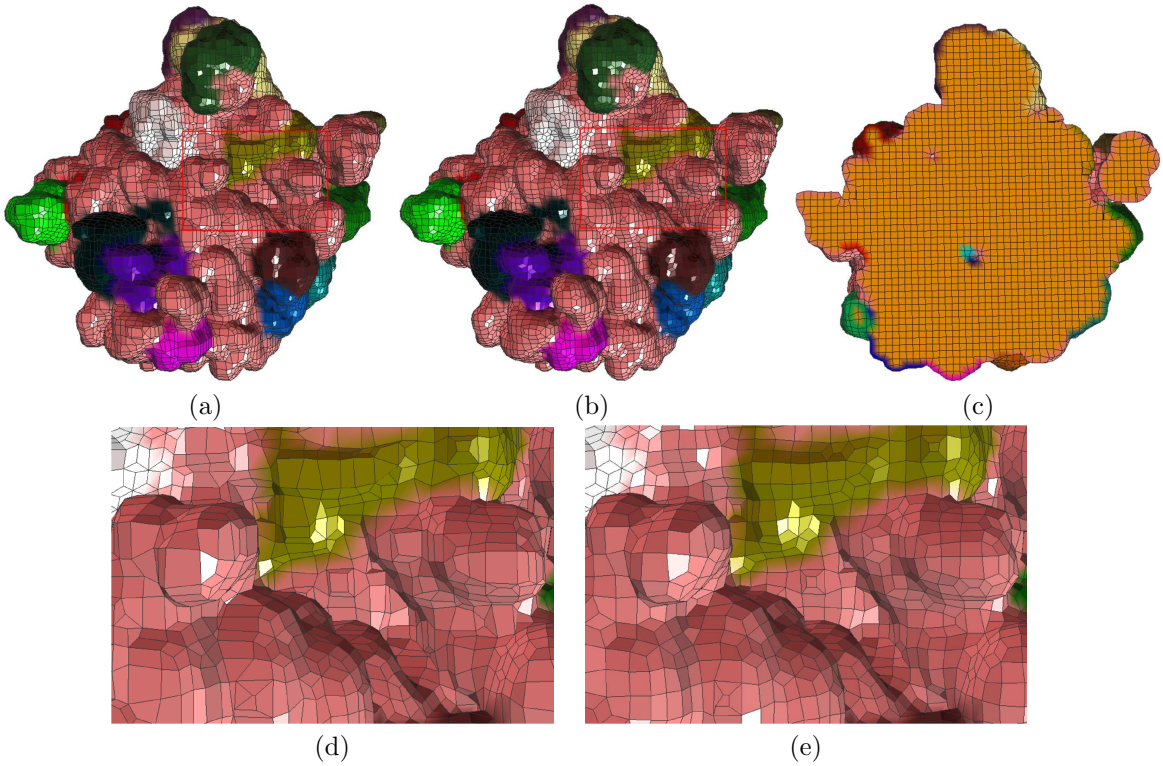


Figure 11. The comparison of mesh quality of Haloarcula Marismortui large Ribosome 50S (1JJ2) crystal subunit. The light yellow and the pink color show 5S and 23S rRNA respectively, the remaining colors are proteins. (a) the original quad mesh (17278 vertices, 17328 quads); (b) the improved quad mesh; (c) the improved hex mesh (57144 vertices, 48405 hexes); (d) the zoom-in picture of the red box in (a); (e) the zoom-in picture of the red box in (b).

We have applied our surface smoothing and quality improvement technique on some biomolecular meshes. In Figure 4, the surface of a molecule consisting of three amino acids is denoised, the surface quadrilateral mesh becomes more regular and the aspect ratio is improved. The comparison of the quality of quad/hex meshes of Ribosome 30S/50S are shown in Figure 1, Figure 11 and Figure 10. The surface diffusion flow preserves a sphere accurately when the initial mesh is embedded and close to a sphere and the tangent movement of boundary vertices doesn't change the shape, therefore features on the molecular surface are preserved. Our quality improvement scheme also works for adaptive meshes as shown in Figure 7.

From Figure 6 and 8, we can observe that the mesh, especially the surface mesh, becomes more regular and facial features of the human head are preserved as well as the aspect ratio is improved (Figure 10). The interior and exterior hexahedral meshes of the human head as shown in Figure 12 have been used in the electromagnetic scattering simulations [36]. Figure 13 shows the quality improvement of hexahedral meshes, as well as the surface quadrilateral meshes, of the human knee and the Venus model. Figure 14 and Figure 10 show the comparison of quadrilateral meshes of the visual human (male), which are generated from a CT data.

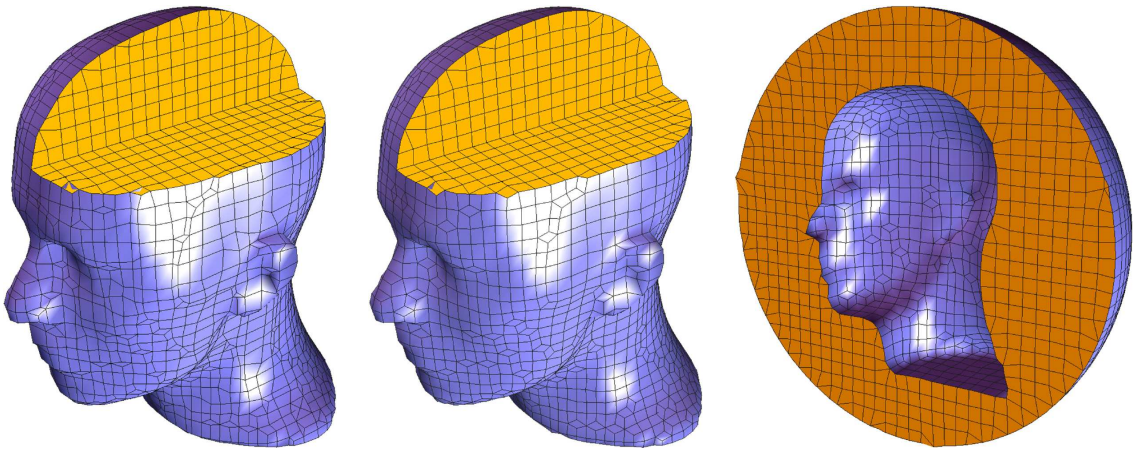


Figure 12. The comparison of mesh quality of the interior and exterior hexahedral meshes. (a) the original interior hex mesh (8128 vertices, 6587 hexes); (b) the improved interior hex mesh; (c) the improved exterior hex mesh (16521 vertices, 13552 hexes). The mesh quality is measured by three quality metrics as shown in Figure 10.

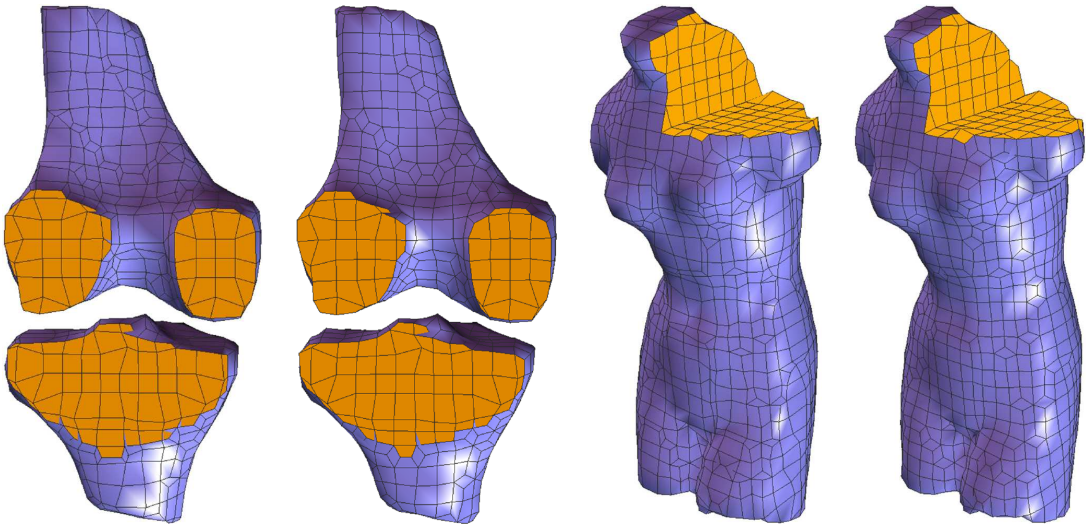


Figure 13. The comparison of mesh quality of the human knee and the Venus model. (a) the original hex mesh of the knee (2103 vertices, 1341 hexes); (b) the improved hex mesh of the knee; (c) the original hex mesh of Venus (2983 vertices, 2135 hexes); (d) the improved hex mesh of Venus.

6. CONCLUSIONS

We have presented an approach to smooth the surface and improve the quality of quadrilateral and hexahedral meshes. The surface diffusion flow is selected to denoise surface

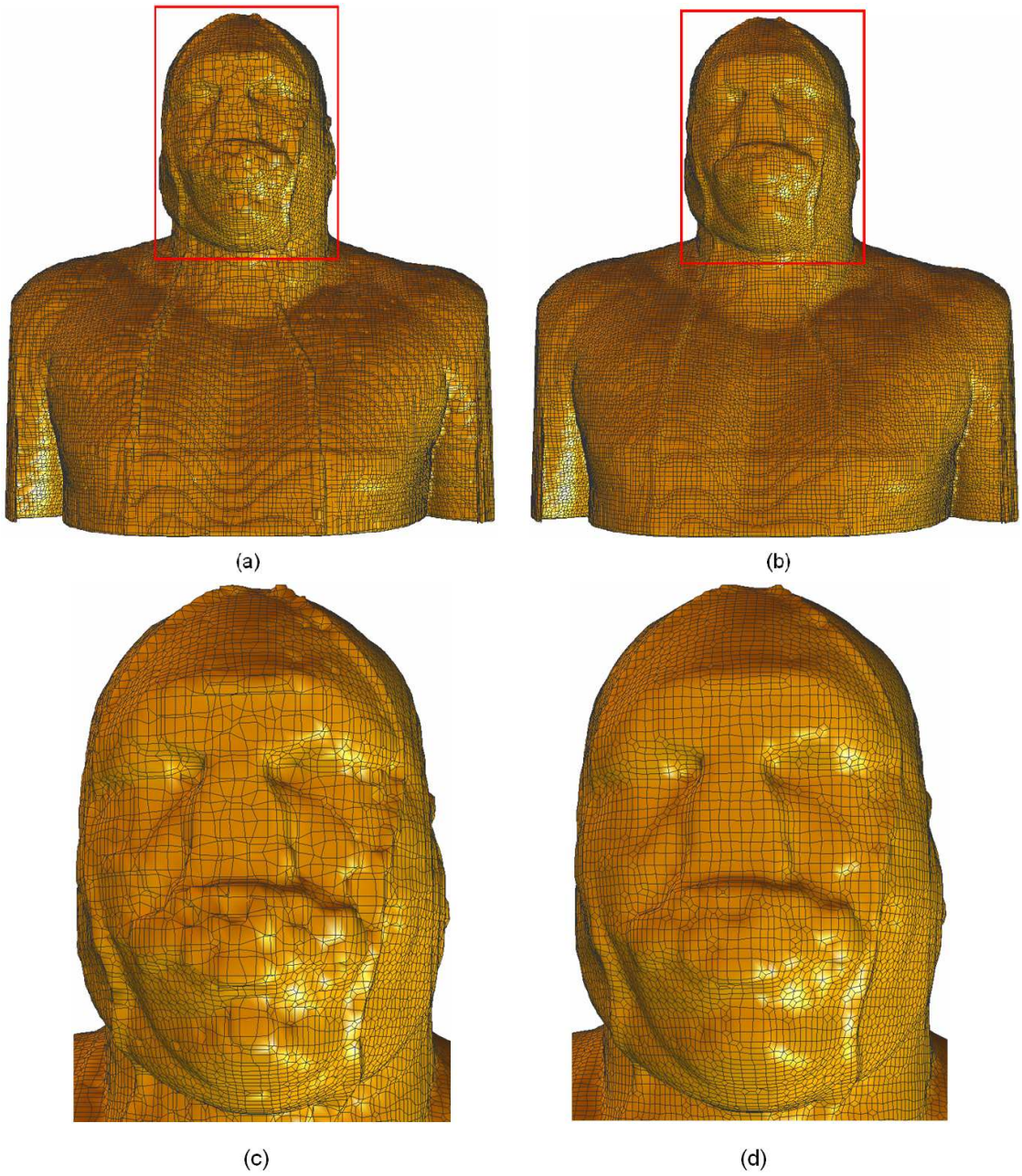


Figure 14. The comparison of mesh quality of the visual human (male). (a) the original quad mesh (76975 vertices, 77708 quads); (b) the improved quad mesh; (c) the zoom-in picture of the red box in (a); (d) the zoom-in picture of the red box in (b).

meshes by adjusting each boundary vertex along its normal direction. The surface diffusion flow is volume preserving, and also preserves a sphere accurately when the input mesh is embedded and close to a sphere, therefore it is especially suitable for surface smoothing of biomolecular meshes because biomolecules are usually modelled as a union of hard spheres. The vertex tangent movement doesn't change the surface shape, therefore surface features can be preserved. The interior vertices of hex meshes are relocated to their mass centers in order to improve the aspect ratio. In a summary, our approach has the properties of noise removal, feature preservation and mesh quality improvement. The resulting meshes are extensively used for efficient and accurate finite element calculations.

We have developed an interactive program for surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow, and plugged it into our LBIE-Mesh software (Level Set Boundary and Interior-Exterior Mesher), which can generate adaptive and quality 2D (triangular/ quadrilateral) and 3D (tetrahedral/hexahedral) meshes from volumetric data. The algorithm of tetrahedral mesh generation is described in [2], and the algorithm of quadrilateral/hexahedral mesh generation is described in [1]. Our results were computed on a PC equipped with a Pentium III 800MHz processor and 1GB main memory.

ACKNOWLEDGEMENTS

An early version of this paper appeared in 14th International Meshing Roundtable conference [37]. The work at University of Texas was supported in part by NSF-ITR grants ACI-0220037, EIA-0325550 and grants from the NIH 0P20 RR020647, R01-GM074258, and R01-9GM073087. A substantial part of the work on this paper was done when Prof. Guoliang Xu was visiting Prof. Chandrajit Bajaj at UT-CVC. His work was partially supported by the aforementioned grants, the J.T. Oden ICES fellowship and in part by National Science Foundation of China grant (10371130), National Key Basic Research Project of China (2004CB318000).

REFERENCES

1. Zhang Y, Bajaj C. Adaptive and Quality Quadrilateral/Hexahedral Meshing from Volumetric Data. *Computer Methods in Applied Mechanics and Engineering (CMAME)* 2006; **195**(9-12):942–960.
2. Zhang Y, Bajaj C, Sohn B-S. 3D Finite Element Meshing from Imaging Data. *The special issue of Computer Methods in Applied Mechanics and Engineering (CMAME) on Unstructured Mesh Generation* 2005; **194**(48-49):5083–5106.
3. Meyer M, Desbrun M, Schröder P, Burr A. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. *VisMath'02, Berlin* 2002.
4. Xu G, Pan Q, Bajaj C. Discrete Surface Modelling Using Partial Differential Equations. *Computer Aided Geometric Design, accepted* 2005.
5. Xu G. Discrete Laplace-Beltrami Operators and Their Convergence. *Computer Aided Geometric Design* 2004; **21**:767–784.
6. Sapiro G. *Geometric Partial Differential Equations and Image Analysis* Cambridge, University Press, 2001.
7. Canann S, Tristano J, Staten M. An Approach to Combined Laplacian and Optimization-based Smoothing for Triangular, Quadrilateral and Quad-dominant Meshes. *7th International Meshing Roundtable* 1998; 479–494.
8. Freitag L. On Combining Laplacian and Optimization-based Mesh Smoothing Techniqes. *AMD-Vol. 220 Trends in Unstructured Mesh Generation* 1997; 37–43.
9. Owen S. A survey of Unstructured Mesh Generation Technology. *7th International Meshing Roundtable* 1998.
10. Teng S-H, Wong CW. Unstructured Mesh Generation: Theory, Practice, and Perspectives. *International Journal of Computational Geometry and Applications* 2000; **10**(3):227–266.

11. Field D. Laplacian Smoothing and Delaunay Triangulations. *Communications in Applied Numerical Methods* 1988; **4**:709–712.
12. George PL, Borouchaki H. *Delaunay Triangulation and Meshing, Application to Finite Elements* 1998; 230–234.
13. Zhou T, Shimada K. An Angle-Based Approach to Two-Dimensional Mesh Smoothing. *9th International Meshing Roundtable* 2000; 373–384.
14. Shontz SM, Vavasis SA. A Mesh Warping Algorithm Based on Weighted Laplacian Smoothing. *12th International Meshing Roundtable* 2003; 147–158.
15. Knupp P. Winslow Smoothing on Two Dimensinoal Unstructured Meshes. *Engineering with Computers* 1999; **5**:263–268.
16. Freitag L, Plassmann P. Local Optimization-based Simplicial Mesh Untangling and Improvement. *International Journal for Numerical Method in Engineering* 2000; **49**:109–125.
17. Charalambous C, Conn A. An Efficient Method to Solve the Minimax Problem Directly. *SIAM Journal of Numerical Analysis* 1978; **15**(1):162–187.
18. Freitag L, Ollivier-Gooch C. Tetrahedral Mesh Improvement using Swapping and Smoothing. *International Journal Numerical Methathemical Engineering* 1997; **40**:3979–4002.
19. Lohner R, Morgan K, Zienkiewicz OC. Adaptive Grid Refinement for Compressible Euler Equations. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations, I. Babuska et. al. eds* Wiley, 1997; 281–297.
20. Shimada K, Yamada A, Itoh T. Anisotropic triangular meshing of parametric surfces via close packing of ellipsoidal bubbles. *6th International Meshing Roundtable* 1997; 375–390.
21. Bossen FJ, Heckbert PS. A pliant method for anisotropic mesh generation. *5th International Meshing Roundtable* 1996; 63–76.
22. Baker T. Identification and Preservation of Surface Features. *13th International Meshing Roundtable* 2004; 299–310.
23. Garimella RV, Shashkov MJ, Knupp PM. Optimization of Surface Mesh Quality Using Local Parametrization. *11th International Meshing Roundtable* 2002; 41–52.
24. Semenova I, Savchenko V, Hagiwara I. Two Techniques to Improve Mesh Quality and Preserve Surface Characteristics. *13th International Meshing Roundtable* 2004; 277–288.
25. Staten M, Canann S. Post Refinement Element Shape Improvement for Quadrilateral Meshes. *AMD-Trends in Unstructured Mesh Generation* 1997; **220**:9–16.
26. Kinney P. CleanUp: Improving Quadrilateral Finite Element Meshes. *6th International Meshing Roundtable* 1997; 437–447.
27. Mitchell S, Tautges T. Pillowing Doublets: Refining a Mesh to Ensure That Faces Share at Most One Edge. *4th International Meshing Roundtable* 1995; 231–240.
28. Schneiders R. Refining Quadrilateral and Hexahedral Element Meshes. *5th International Conference on Grid Generation in Computational Field Simulations* 1996; 679–688.
29. Bajaj C, Warren J, Xu G. A Subdivision Scheme for Hexahedral Meshes. *The Visual Computer* 2002; **18**(5-6):343–356.
30. Escher J, Mayer UF, Simonett G. The Surface Diffusion Flow for Immersed Hypersurfaces. *SIAM Journal of Mathematical Analysis* 1998; **29**(6):1419–1433.
31. Willmore TJ. *Riemannian Geometry* Glareden Press, Oxford, 1993.
32. Oddy A, Goldak J, McDill M, Bibby M. A Distortion Metric for Isoparametric Finite Elements. *Transactions of CSME, No. 38-CSME-32, Accession No. 2161* 1988.
33. Knupp P. Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities. Part I - A Framework for Surface Mesh Optimization. *International Journal Numerical Methathemical Engineering* 2000; **48**:401–420.
34. Knupp P. Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities. Part II - A Framework for Volume Mesh Optimization and the Condition Number of the Jacobian Matrix. *International Journal Numerical Methathemical Engineering* 2000; **48**:1165–1185.
35. Kober C, Matthias M. Hexahedral Mesh Generation for the Simulation of the Human Mandible. *9th International Meshing Roundtable* 2000; 423–434.
36. Xue D, Demkowicz L, Bajaj C. Reconstruction of G^1 Surfaces with Biquartic patches for hp FE Simulations. *13th International Meshing Roundtable* 2004; 323–332.
37. Zhang Y, Bajaj C, Xu G. Surface Smoothing and Quality Improvement of Quadrilateral/Hexahedral Meshes with Geometric Flow. *14th International Meshing Roundtable* 2005; 449–468.