1. Abstract

2. Introduction

2.1. Related works

2.2. Contributions

3. Edge Sensitive Mesh Subdivision

In this section, we provide two methods for triangular mesh subdivision: Edge Sensitive Mesh Subdivision, and Advanced Edge Sensitive Mesh Subdivision. Both methods follow the iterated scheme of subdivision plus smoothing, mentioned in Loop Subdivision[1].

3.1. Edge Sensitive Mesh Subdivision

Loop's method can be expressed in terms of linear subdivision and an averaging scheme to approximate a spherical surface, which will efface edges and vertices features. The phenomenon is common when applying Catmull method[2], and Butterfly method[3]. When the mesh is rigid, these iterations will fail more or less as illustrated in Figure 1.

a b c d e

Figure 1 a) original mesh. b) Loop Subdivision. c) Catmull Subdivision. d) Butterfly Subdivision. e) Edge Sensitive Mesh Subdivision. Except a), all these subdivisions have been applied five times on the original mesh.

a b c d e

Figure 2 a) original mesh. b) Loop Subdivision. c) Catmull Subdivision. d) Butterfly Subdivision. e) Edge Sensitive Mesh Subdivision. Except a), all these subdivisions have been applied five times on the original mesh.

Our proposed Edge Sensitive Mesh Subdivision method implements linear one to four triangular mesh subdivisions illustrated in Figure 3, and smoothed by proposed smooth operator which will generate an edge sensitive result.

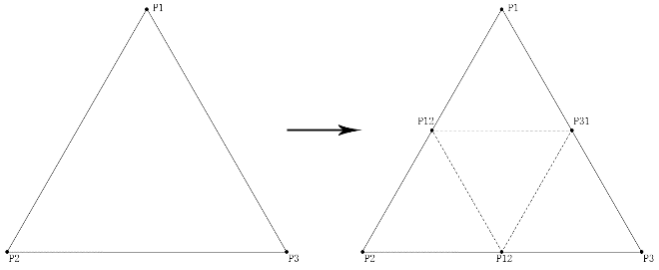


Figure 3 Linear 1 to 4 subdivision of triangles for triangular subdivision schemes.

Let's denote the original mesh as $\mathbf{M}_i$. To perform linear triangular subdivision, we insert middle vertices on the edge of each triangle to the hash map $H_n(v_i,h_i)$ for storing vertex coordination v_i and its handle h_i as corresponding hash key. For each vertex on mesh $\{v_1,v_2,\dots,v_n\}$, check if the generated middle edge point v_k is already in the map. If so, get its handle h_k for the on-coming face generation, else, insert the point v_k into the mesh and create its handle h_k while update vertices hash map $H_{n+1}(v_i,h_i)$. Finally, form the new triangular surfaces using vertex handles geometrically anticlockwise, and eliminate elder redundant faces simultaneously. Each triangle will then be split into four sub-triangles and original mesh $\mathbf{M}_i$ is subdivided to $\mathbf{M}_{i+1}$.

Smoothing for triangular meshes will be applied on not only the previous vertices on $\mathbf{M}_i$ but all vertices on the generated mesh $\mathbf{M}_{i+1}$ whose two-ring vertex set is derived from one-ring vertices of previous mesh $\mathbf{M}_i$ as illustrated in Figure 4. For each vertex on $\mathbf{M}_{i+1}$, we use a one-ring neighbor weighted centroid method for vertices as shown in Figure 5. The weight of each one-ring neighbor β is decided by the number of one-ring neighbors n.

$$\beta = \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2$$

Equation 1

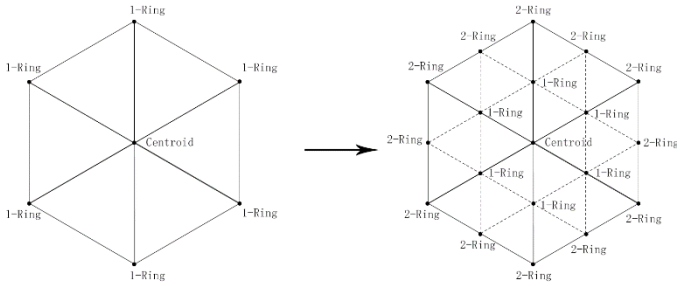


Figure 4 One-ring vertices Derivative

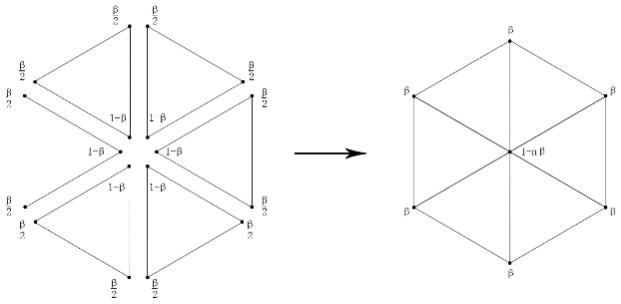
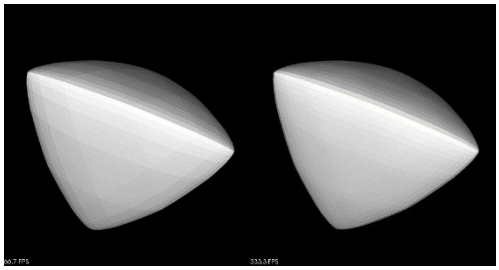


Figure 5 One-ring neighbor weighted centroid method for triangular vertices

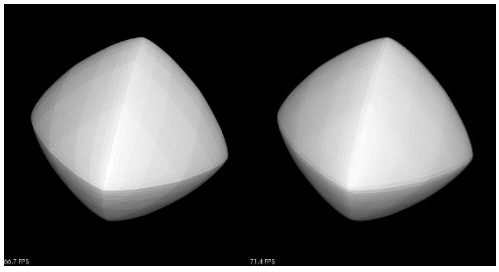
3.2. Advanced Edge Sensitive Mesh Subdivision

By using Edge Sensitive Mesh Subdivision, an edge sensitive result is generated. But the method will generate model dependent noise waves on the mesh, as illustrated in Figure 6.



a b

Figure 6 a) Model dependent noise waves on the mesh. b) Advanced Edge Sensitive Mesh Subdivision. Both subdivisions have been applied five times on the original mesh.



a b

Figure 7 a) Model dependent noise waves on the mesh. b) Advanced Edge Sensitive Mesh Subdivision. Both subdivisions have been applied five times on the original mesh.

We propose an Advanced Edge Sensitive Mesh Subdivision which is a combination of

Edge Sensitive Mesh Subdivision and Loop Method. The Subdivision scheme consists of several iterations of Edge Sensitive Subdivision and an iteration of Loop Subdivision at the end of algorithm. The implementation resembles the Edge Sensitive Subdivision, except the generation of edge points and smoothing operator.

Unlike the middle edge points generation in Edge Sensitive Subdivision, the edge points in the last iteration of Advanced Edge Sensitive Subdivision are generated based on weighted neighbors illustrated in Figure 8, where directly connected vertices take up a greater weight, say 3/8 each, and that indirect weight slightly less, say 1/8 each.

Neighbor vertices for smoothing method depends on previous one-ring vertices on M_i but not newly generated vertices on M_{i+1}. The weight of each neighbors follows the same principle introduced in Edge Sensitive Subdivision.

4. Keypoints-based Edge Sensitive Mesh Subdivision

The result of subdivision should depend on the feature of meshes, which will lead a better subdivision result and higher efficiency. In this section, we propose a detection of imbalanced vertices in 3D meshes, and using detected keypoints to guide subdivision procedure.

4.1. Imbalanced Vertices Detection

We first propose a vertex geometric feature based operator, which is a two-dimensional implementation of keypoints selection. We extend Li's [4] previous work on imbalanced keypoints detection to triangular meshes.

The basic idea is using projections to transform 3D geometric features to two-dimensional spaces. Let $\mathrm{M}(\mathrm{V},\mathrm{F},\mathrm{N}_F)$ be a triangular mesh where V is the set of vertices, F is the set of faces, and N_F represents the set of face normals. Suppose a projection P_T will project any vector to the plane T, 3D geometric mesh features will be project to two-dimension when T is the tangent plane of mesh vertices V.

Imbalanced point detection in 2D images aims to minimize the occurrences of edge points [4]. Denote I a gray value image, p a local point, $\theta_i = \frac{(i-1)2\pi}{N}$, and $\mathrm{l}_i = (\cos\theta_i, \sin\theta_i)$ for $\mathrm{i} = 1, 2, \ldots, \mathrm{N}$. Denote $\frac{\partial \mathrm{I}}{\partial \mathrm{l}_i}(p)$ a directional derivative of p along l_i direction. We cluster $\left\{\frac{\partial \mathrm{I}}{\partial \mathrm{l}_i}(p)\right\}_{i=1}^{N}$ into two classes in terms of their magnitudes $\left|\frac{\partial \mathrm{I}}{\partial \mathrm{l}_i}(p)\right|$. If two clusters have the same size, the image point p is balanced.

The sorting method proposed in [4] to classify $\left\{\frac{\partial \mathrm{I}}{\partial \mathrm{l}_i}(p)\right\}_{i=1}^{N}$ can be generalized to extract 3D imbalanced vertices with normal vector projections. Let $\mathrm{maxDiff}$ be the max difference and D be the index of maximum difference:

$$\mathrm{maxDiff} = \max_j(\beta_{j+1} - \beta_j)$$

$$\mathrm{D} = \operatorname{argmax}_j(\beta_{j+1} - \beta_j)$$

Where β represents projected vectors, and $1 \le \mathrm{j} \le \mathrm{N} - 1$. Given a threshold on homogeneity ε, the imbalanced vertex can be defined under the condition that $\mathrm{maxDiff} < \varepsilon$:

$$\mathrm{IMB}(\mathrm{v}_i) = \begin{cases} 1 & D_i < \dfrac{N}{2} \\ 0 & else \end{cases}$$

4.2. Keypoints based Edge Sensitive Mesh Subdivision

Based on imbalanced keypoints detection, all the vertices are classified into two categories. According to the properties of imbalanced vertices, our keypoints distributed mainly at the boundaries and detailed parts, which should be remained after subdivision but slightly adjusted with neighbors. We propose a subdivision scheme, which contains edge point generation and smooth operator, to balance the emphasis of original features and smooth of subdivided mesh.

The unit of edge point generation in our provided method is a couple of triangles back to back as illustrated in Figure 8. Denote the edge point generator in Edge Sensitive Subdivision and Advanced Edge Sensitive Subdivision as $G_{ESS}(N)$ and $G_{AESS}(N)$ respectively, where N is the set of neighbor points.

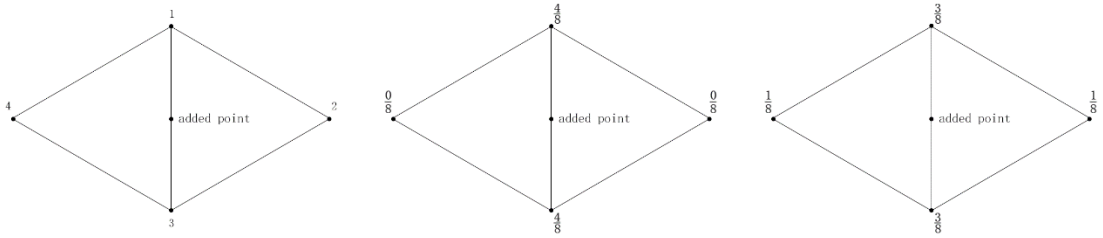


Figure 8 Unit of edge point generation (left) Edge Sensitive Subdivision edge point generator (middle). Advanced Edge Sensitive Subdivision edge point generator (right)

The balance can be guided by the amount of imbalanced keypoints in an isolated generation unit and implemented by a weighted combination of $G_{ESS}(N)$ and $G_{AESS}(N)$. For a precise and accurate calculation, the weight of two generators should be geometrically related and geometrically symmetric. The weight of $G_{ESS}(N)$ and $G_{AESS}(N)$ will be defined as $\mathrm{W}_{G_{ESS}}$ and $\mathrm{W}_{G_{AESS}}$, which depend on the number of keypoints in a generation unit, and obey the following principles.

$$\mathrm{W}_{G_{ESS}} = \sum_{i=1}^{|N|} isKeypoint(N_i)$$

$$\mathrm{W}_{G_{AESS}} = |N| - \sum_{i=1}^{|N|} isKeypoint(N_i)$$

$$\text{isKeypoint}(N_i) = \begin{cases} 1 & N_i \ is \ a \ keypoint \\ 0 & N_i \ isn't \ a \ keypoint \end{cases}$$

A combination of two edge point generator $G(N)$ will be defined

$$G(N) = \mathrm{W}_{G_{ESS}} G_{ESS}(N) + \mathrm{W}_{G_{AESS}} G_{AESS}(N)$$

Smooth operator will also be guided by keypoints, which extends the previous smooth operator. Neighbor vertices for smooth operator will be different which depends on whether the centroid vertex is a keypoint or not. If so, neighbor vertices should be elder one-ring vertices on M_i, else, one-ring vertices on M_{i+1}.

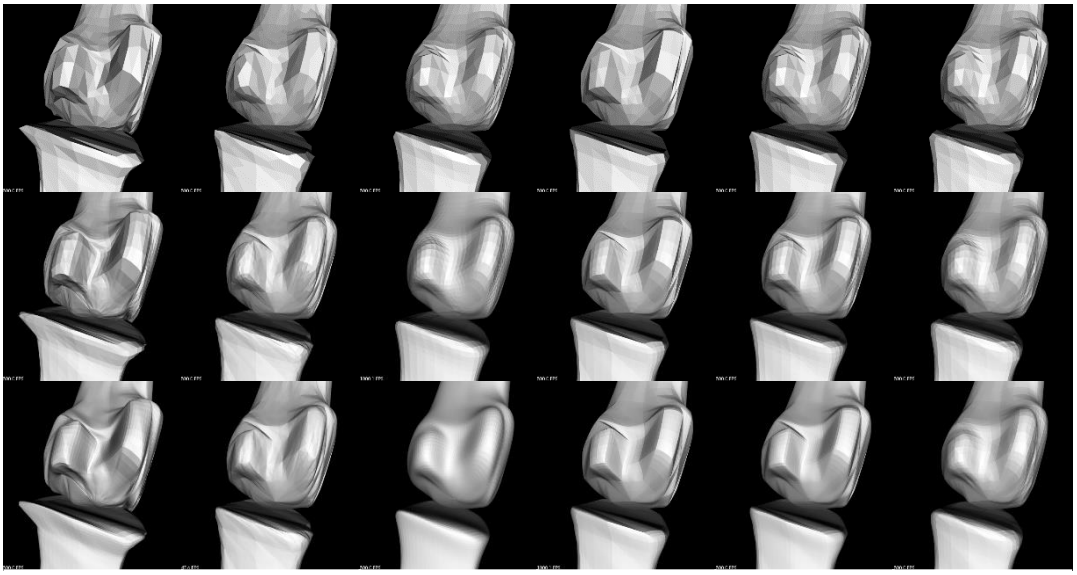
5. Experiment

We test different subdivision methods on several biomedical data, including phalanx (distal phalanx, middle phalanx, and proximal phalanx), cuneiform bone (intermediate cuneiform bones, entocuneiform, ectocuneiform), central anlrle bone, cuboid bone and so on provided by Florida State University.

As the time of subdivision methods iteration increases, the performance of different algorithm is becoming easier to differentiate, as illustrated in Figure 9. Except Catmull method, all the subdivision methods implement one to four triangular faces subdivision, which means each mesh in a row has the same number of vertices and faces, except Catmull subdivision. The number of vertices and triangular faces in each iteration follows the principle when apply one to four mesh subdivision schemes.

$$V(M_i) = V(M_{i-1}) + \frac{3}{2} F(M_{i-1})$$

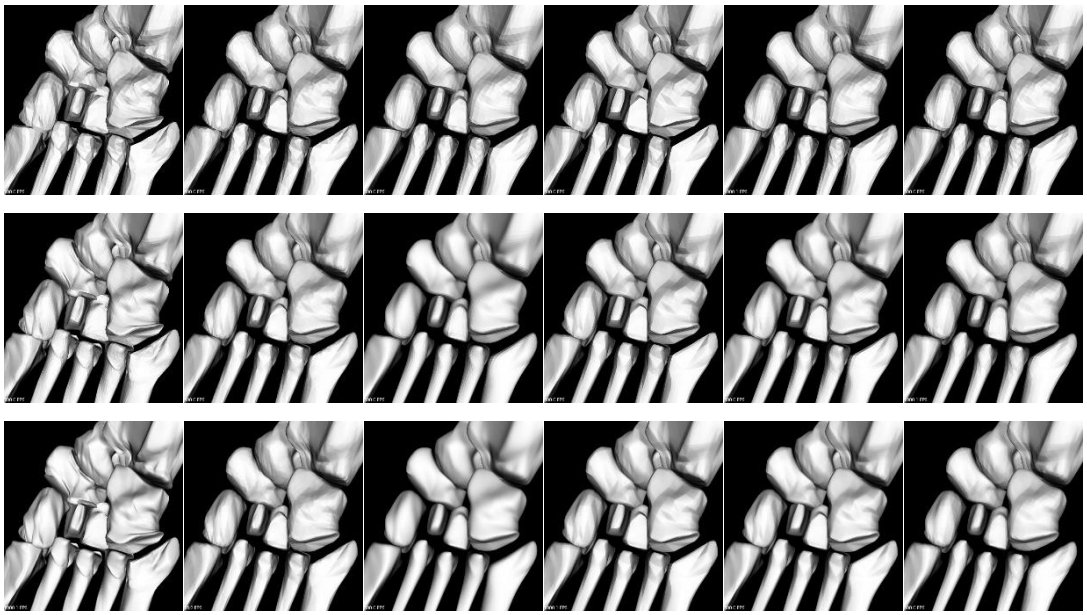
$$F(M_i) = 4V(M_{i-1})$$

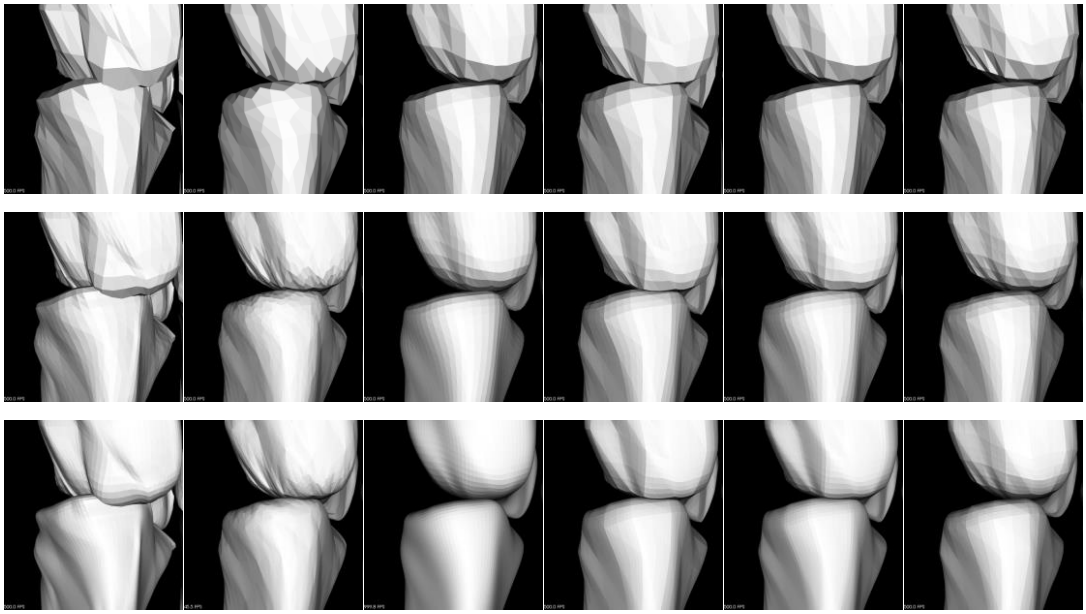


a1	b1	c1	d1	e1	f1
a2	b2	c2	d2	e2	f2
a3	b3	c3	d3	e3	f3

Figure 9 Subdivision Comparison on same biomedical data. a) Butterfly Subdivision. b) Catmull Subdivision. c) Loop Subdivision. d) Edge Sensitive Subdivision. e) Advanced Edge Sensitive Subdivision. f) Keypoints-based Edge Sensitive Subdivision. Iteration time increase by row.

Obviously, subdivision result after Loop method has a better spherical resembling appearance, while Edge Sensitive Subdivision provide a better property of edge feature remaining. Butterfly subdivision leads mesh to a more boundary malleable shape, while Catmull present more vertex details. Advanced Edge Sensitive Subdivision accomplish an edge feature remaining but smooth result, but compared with Keypoint-based Edge Sensitive Subdivision, the later algorithm provides a better result.





6. Reference

[1] CT. Loop. Smooth subdivision surfaces based on triangles. Masters thesis. University of Utah, Dept. of Mathematics. 1987.

[2] E. Catmull, and J Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design. 1978.

[3] Butterfly

[4] Q. Li, J. Ye, and C. Kambhamettu. Interest point detection using imbalance oriented selection. Pattern Recognition,

[5] K. Feng, Q. Li, Y. Gong, J. Yang. Detection of imbalanced vertices in 3D meshes.

[6]