

# A Feature Preserved Mesh Subdivision Framework for Biomedical Mesh

Jing Yang, Yongyi Gong, Hefeng Wu  
*School of Information Science and Technology  
Guangdong University of Foreign Studies  
Guangzhou, China  
jingyang\_carl@qq.com*

Qi Li  
*Department of Computer Science  
Western Kentucky University  
Bowling Green, KY, USA*

**Abstract**—As biomedical data in 3D space collected increasingly, there is a pressing need for efficient and accurate applications in the field of bioinformation analysis. For biomedical purpose, mesh subdivision techniques are commonly used to generate adaptive multi-resolution meshes for fast or accurate algorithms. However, current smoothing methods for each subdivision algorithm will moderate edge and vertex features from the original mesh. In this paper, we propose a feature preserved mesh subdivision framework, which generates a visually sensitive and a more precise result compared with commonly used subdivision methods, to preserve edge and vertex geometrical features of biomedical data.

**Keywords**-mesh subdivision, keypoints, feature preserved, biomedical mesh

## I. INTRODUCTION

Recent years, 3D computer graphical technology works diffusely as biomedical assistants, including utilizations of three dimensional CT (3D CT) reconstructions [1]–[4], presentations of skeletons and tissues [5], and comparisons of genome [6]–[9], and etc., due to its visual obviousness. The technique is widely used for presenting facial bones and their connections, confirmation of borderline or size of diseased tissues and their relation to the adjacent tissues. With the aid of bio-images and bio-models, diagnosis and operations on clinical medicine accomplished a more accurate precision. Moreover, substantial effort has been made around world to determine spatial expression patterns of genes in mammalian genome using experimental techniques such as *in situ* hybridization(ISH) [10]. Performing ISH on multiple subjects yields expression images of various genes over the common anatomical structure and comparing these images reveals the spatial relations between genes, which are often key to understanding their functional relations [11] [12].

The pros of using Computer Graphics including 3D spatial and triangular meshes [13]–[15] to expose nicety of bioinformation has created an ever-titanic amount of spatial data (in the form of 2D medical images, and 3D bio-models) which requires efficient and accurate process and analysis consequently. The anatomical differentiations among these biomedical models aggravate the computational challenges involved in visual and medical comparison among data collected from different modes [16].

Restrained by the calculation performance of physical devices, medical detailed expression and accuracy of biomedical data is circumscribed. Solutions emerge as medical processing required, one of which is using meshes without redundant vertices and faces. The contradiction of biomedical precision and data optimization will generate as model shrinking. Hence, feature-based mesh subdivision present state-of-the-art results [16].

Using feature-based subdivision to suitably amplify medical precision and organize spatial data into multi-resolution versions, on which visual comparison and diagnosis will have a higher performance [16]. Meanwhile, the multi-resolution structure of a subdivision mesh further gives rise to fast algorithms for processing and accurate comparison for tiny bio-divergence.

Commonly used mesh subdivision methods including Butterfly method [17], Loop subdivision [18], and Catmull-Clark subdivision [19], whose experimental results will be illustrated in Fig.1. Moreover, Xie [20] provide a solid-shell element based triangular subdivision to avoid interpenetration. Liu [21] introduced a dense reconstruction algorithm for mesh subdivision. Amresh [22] developed a subdivision scheme derived from Loop scheme and using watershed segmentation. Dirc Rose [23] proposed an adaptive process that stores the next splitting vertex and temporary triangle based on Modified Butterfly scheme. Kobbelt [24] developed refinement for both his Kobbelt scheme and newly introduced  $\sqrt{3}$  subdivision. Seeger [25] introduced a subdivision scheme based on Butterfly Scheme using quark. In their method, subdivision is controlled by the faces of the original mesh and mesh features are not suitably preserved.

In this paper, we propose a feature preserved mesh subdivision framework to maintain geometrical features, include edges and imbalanced vertices, of original biomedical meshes while give out a more adaptive mesh derivative. Our contributions include a) imbalanced keypoint detection for point feature preservation, b) edge feature preserved mesh subdivision for edge feature preservation, c) model-dependent wavelike noise elimination for result optimization. Subdivision result will be compared after same time of iterations and results show the promise of proposed method on feature maintained mesh subdivisions.

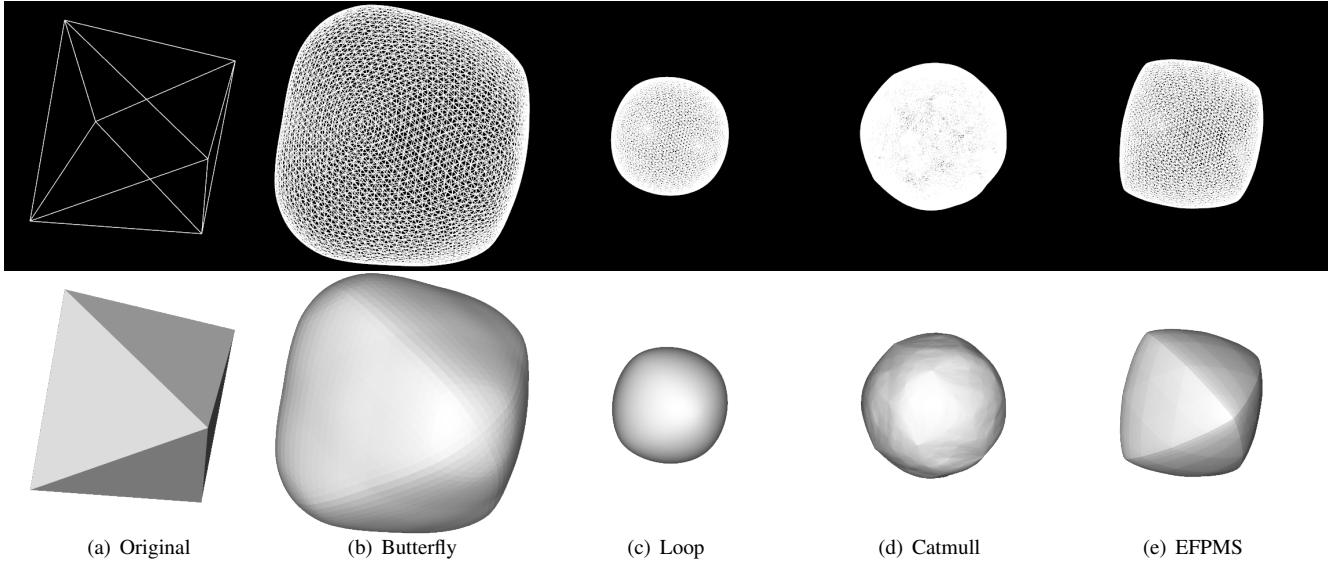


Figure 1: a) Original Mesh. b) Butterfly Subdivision. c) Loop Subdivision. d) Catmull Subdivision. e) Edge Feature Preserved Mesh Subdivision (implementation of our framework’s core module). All subdivision algorithms have been applied five times on original mesh.

## II. COUPLING MODULES OF SUBDIVISION FRAMEWORK: FUNDAMENTAL SUBALGORITHMS

In this section, we will give an overview of algorithm modules within our proposed framework where Imbalanced Keypoint Detection focuses on vertex features while Edge Feature Preserved Mesh Subdivision (EFPMS) underscores edge features. A brief introduction on imbalanced-vertex based feature detection will first be exhibited and then provide our algorithm kernel, EFPMS, while propose a corresponding optimization solution to eliminate model-dependent wavelike noise generated from that. Moreover, the kernel module follows subdivision scheme composed of linear triangular face subdivision and vertex smoothing.

### A. Imbalanced Keypoint Detection

We first propose a geometric feature-based vertex operator, which is a three-dimensional implementation of keypoints detection, to pick up imbalanced keypoints. We extend Li’s [26] previous work on imbalanced keypoints detection to triangular meshes.

The basic idea of vertex operator is using projections to transform 3D geometric features to two-dimensional space. Let  $M(V, F, N_F)$  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 onto the plane  $T$ , 3D geometric mesh features will be transformed to two dimensions when  $T$  is the tangent plane of the given mesh vertex in vertex set  $V$ . Face normals  $N_F$  will then be transformed into its projected vector set  $N_{F'}$ . Finally, we settle all normalized vectors in  $N_{F'}$  to a

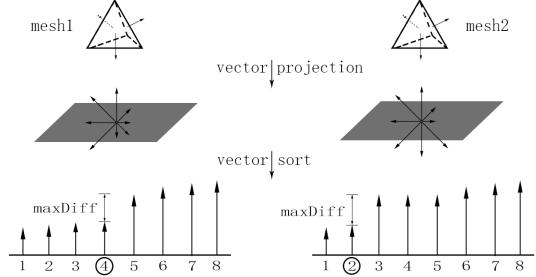


Figure 2: Illustration of imbalance selection. After projecting face normals around vertex on its tangent plane, arrows (suppose 8) are sorted in terms of their magnitudes. L) balanced edge point, where index of maximum difference is 4. R) imbalanced edge point, where that is 2.

polar coordinate system and calculate the set of cross angles between each vector side by side, denote as  $A_F$ .

Imbalanced point selection in 2D images aims to minimize the occurrences of edge points [26] as illustrated in Fig.2. Denote  $I$  a gray value image,  $p$  a local point,  $\theta_i = \frac{(i-1)2\pi}{N}$ , and  $l_i = (\cos \theta_i, \sin \theta_i)$  for  $i = 1, 2, , N$ . Denote  $\frac{\partial I}{\partial l_i}(p)$  a directional derivative of  $p$  along  $l_i$  direction. We cluster  $\{\frac{\partial I}{\partial l_i}(p)\}_{i=1}^N$  into two classes in terms of their magnitudes  $|\frac{\partial I}{\partial l_i}(p)|$ . If two clusters have the same size, the image point  $p$  is balanced.

The sorting method proposed in Li’s method is to classify  $\{\frac{\partial I}{\partial l_i}(p)\}_{i=1}^N$ , which can be generalized to extract 3D imbalanced vertices with proposed operator. Let  $maxDiff$  be the max difference and  $D$  be the index of maximum difference:

$$maxDiff = \max_j (\alpha_{j+1} - \alpha_j) \quad (1)$$

$$D = \arg \min_j (\alpha_{j+1} - \alpha_j) \quad (2)$$

where  $\alpha$  represents value in  $A_F$ , and  $1 \leq j \leq N - 1$ . Given a threshold on homogeneity  $\varepsilon$ , the imbalanced vertex can be defined under the condition that  $maxDiff < \varepsilon$ :

$$IMB(v_i) = \begin{cases} 1 & D_i < \frac{N}{2} \\ 0 & \text{else} \end{cases} \quad (3)$$

One of the implementation results is presented in Fig.3 where imbalanced keypoints delineate vertex features.

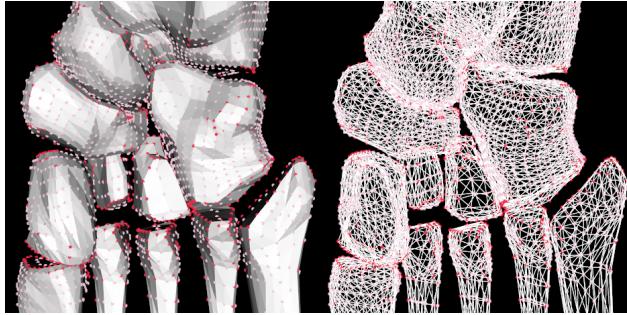


Figure 3: Imbalanced keypoints on footbone mesh with white points representing normal points and red for imbalanced points. L) model view. R) mesh view.

### B. Edge Feature Preserved Mesh Subdivision

Loops method [18] can be expressed as linear subdivision and an averaging scheme to approximate a spherical surface, which will efface edge and vertex features. The phenomenon is common when applying Catmull method, and Butterfly method. When the mesh is rigid, these iterations will fail to preserve the original geometrical features as illustrated in Fig.1. To moderate the feature friction after subdivision and highlight original edge and vertex features, we provide an Edge Feature Preserved Mesh Subdivision (EFPMS) to generate feature adaptive results, where pseudocode is arranged in Algorithm 1.

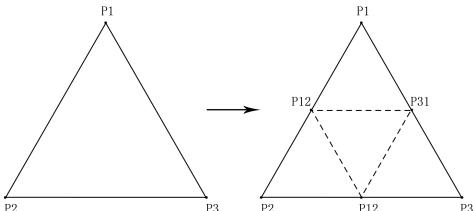


Figure 4: Linear one to four subdivision for triangular subdivisions.

Our proposed method implements linear one to four triangular mesh subdivisions to increase the details in a mesh as illustrated in Fig.4, and smooths by proposed smoothing algorithm to accomplish surface approximation after subdivision while actualize shape retention.

Denote the original mesh as  $M_i$ , and edge point generator  $G_{EFPMS}$  as illustrated in Fig.8. To perform linear triangular subdivision, we insert points created by generator  $G_{EFPMS}$  on the edge of each triangle to the hash map  $H_n(v_i, h_i)$  for storing vertices coordination  $v_i$  and their handles  $h_i$  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  $M_i$  is subdivided to  $M_{i+1}$ .

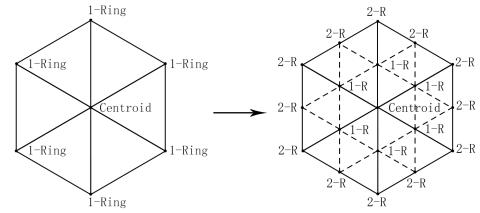


Figure 5: One-ring vertices derivative.

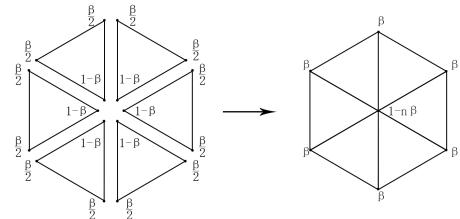


Figure 6: One-ring neighbor weighted centroid method for triangular vertices.

Smoothing for triangular meshes will be applied on not only the previous vertices on  $M_i$  but all vertices on the generated mesh  $M_{i+1}$  whose two-ring vertex set is derived from one-ring vertices of previous mesh  $M_i$  as illustrated in Fig.5. For each vertex on  $M_{i+1}$ , we use a one-ring neighbor weighted centroid method for averaging as shown in Fig.6. The weight of each neighbor  $\beta$  is decided by the number of one-ring neighbors  $n$ .

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

---

**Algorithm 1** Edge Feature Preserved Mesh Subdivision

---

**Input :** Original Mesh  $M_i$   
**Output :** Subdivided Mesh  $M_{i+1}$

```

for each triangular face  $f_i$  in  $M_i$  do
    get 3 edge point on edges of  $f_i$  with  $G_{EFPMS}$ ;
    generate 4 subsfaces anticlockwise;
end for
{NOTE:  $M_i$  is now subdivided into  $M_{i+1}$ }

for each vertex  $v_i$  in  $M_{i+1}$  do
     $N(v_i)$  = 1-ring neighbors of  $v_i$  on current mesh  $M_{i+1}$ ;
     $v_i$  = 1-ring Neighbor Weighted Centroid method on  $v_i$ ;
end for
{NOTE:  $M_{i+1}$  is now smoothed}

```

---

### C. Model-dependent Wavelike Noise Elimination

By using EFPMS, an edge and vertex aware result is generated. But the method will generate model-dependent wavelike noise on the mesh, as illustrated in Fig.7.

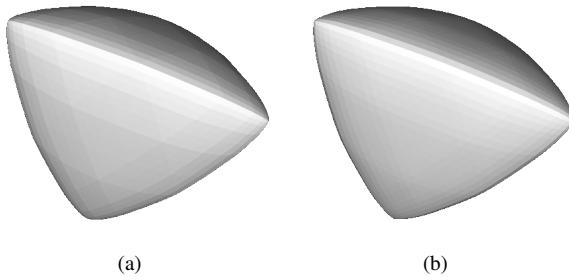


Figure 7: a) Model-dependent wavelike noise. b) Remove wavelike noise by Smoothness-focused Mesh Subdivision. All subdivisions algorithm has been applied five times on original meshes.

We propose Model-dependent Wavelike Noise Elimination, a combination of EFPMS and Smoothness-focused Mesh Subdivision (SMS), to eliminate noise. The Subdivision scheme consists of several iterations of EFPMS and an iteration of SMS at the end of algorithm. The implementation of SMS resembles the EFPMS, except the generation of edge points and smoothing operator.

Unlike edge points generator  $G_{EFPMS}$  in feature preserved subdivision, edge points in the last iteration of noise elimination algorithm are generated based on weighted neighbors illustrated in Fig.8, where directly connected vertices take up a greater weight, say 3/8 each, and indirect weight slightly less, say 1/8 each.

Neighbor vertices for smoothing method depends on pre-

vious one-ring vertices on  $M_i$ , but not newly generated vertices on  $M_{i+1}$ , which is different from the implementation of EFPMS. The weight of each neighbor follows the same principle introduced in EFPMS.

### III. FEATURE PRESERVED MESH SUBDIVISION FRAMEWORK

In this section, we propose a Feature Preserved Mesh Subdivision Framework (FPMS), which contains edge point generation and smooth operator, to balance the emphasis of original features and smoothing extent. The big picture of our framework is using detected points to guide subdivision procedures. Based on imbalanced keypoint detection, all the vertices are classified into two categories and keypoints are distributed mainly at the boundaries and detailed parts, which should be remained after subdivision but slightly adjusted with neighbors.

The unit of edge point generation in our provided method is a couple of triangles back to back as illustrated in Fig.8. Denote edge point generators in Edge Feature Preserved Mesh Subdivision and Smoothness-based Mesh Subdivision as  $G_{EFPMS}(N)$  and  $G_{SMS}(N)$  respectively, where  $N$  is the set of neighbor points.

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

$$W_{G_{EFPMS}} = \sum_{i=1}^{|N|} Key(N_i) \quad (5)$$

$$W_{G_{SMS}} = |N| - \sum_{i=1}^{|N|} Key(N_i) \quad (6)$$

$$Key(N_i) = \begin{cases} 1 & N_i \text{ is a keypoint} \\ 0 & \text{else} \end{cases} \quad (7)$$

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

$$G(N) = W_{G_{EFPMS}} G_{EFPMS}(N) + W_{G_{SMS}} G_{SMS}(N) \quad (8)$$

Smooth operator will also be guided by keypoints, which extends the previous smoothing scheme. 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  $M_{i+1}$ .

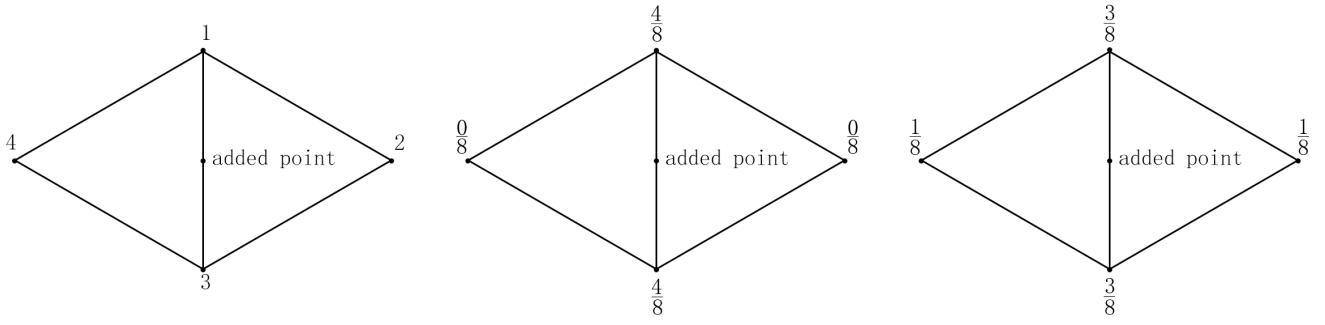


Figure 8: Unit of edge point generator (left). Feature Preserved Mesh Subdivision edge point generator (middle). Smoothness-focused Mesh Subdivision edge point generator (right). Weight of vertices in each generator are tagged aside its position.

---

**Algorithm 2** Feature Preserved Mesh Subdivision

---

**Input :** Original Mesh  $M_i$   
**Output :** Subdivided Mesh  $M_{i+1}$

imbalanced keypoint detection on  $M_i$ ;  
 {NOTE: vertices have been classified into two class}

**for each** triangular face  $f_i$  in  $M_i$  **do**  
 get edge point  $e_{EFPMS}$  on edges of  $f_i$  with  $G_{EFPMS}$ ;  
 get edge point  $e_{SMS}$  on edges of  $f_i$  with  $G_{SMS}$ ;  
 average position of edge points by  $G(N)$ ;  
 generate 4 sub-faces anticlockwise;  
**end for**  
 {NOTE:  $M_i$  is now subdivided into  $M_{i+1}$ }

**for each** vertex  $v_i$  in  $M_{i+1}$  **do**  
**if**  $v_i$  is a keypoint **then**  
 $N(v_i) = 1\text{-ring neighbors of } v_i \text{ on } M_i$ ;  
 $v_i = 1\text{-ring Neighbor Weighted Centroid on } v_i$ ;  
**else**  
 $N(v_i) = 1\text{-ring neighbors of } v_i \text{ on } M_{i+1}$ ;  
 $v_i = 1\text{-ring Neighbor Weighted Centroid on } v_i$ ;  
**end if**  
**end for**  
 {NOTE:  $M_{i+1}$  is now smoothed}

---

#### IV. EXPERIMENT

We test different subdivision methods on several biomedical data, including phalanx, cuneiform bone, central anlre bone, cuboid bone, etc..

As the time of subdivision methods iterated increasingly, the performance of different algorithm is becoming easier to differentiate, as illustrated in Fig.9. Except Catmull method, all the subdivision methods implement one to four triangular 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 following principles when apply one to four mesh subdivision schemes, where  $V(\cdot)$  and  $F(\cdot)$  represent total amount of vertices and faces on a mesh respectively.

$$V(M_{i+1}) = V(M_i) + \frac{3}{2}F(M_i) \quad (9)$$

$$F(M_{i+1}) = 4F(M_i) \quad (10)$$

Obviously, subdivision result after Loop method has a better spherical resembling appearance, while EFPMS provide a better property of edge feature remaining. Butterfly subdivision leads mesh to a more boundary malleable shape, while Catmull present more vertex details. Model-dependent Wavelike Noise Elimination accomplishes an edge feature remaining and smooth result, but compared with FPMS, the later algorithm provides a better result on edge and vertex features. Fig.10 and Fig.11 exhibit more experimental comparisons.

#### V. CONCLUSION

Proposed Edge Feature Preserved Mesh Subdivision method works well on edge features but it will generate troublesome model-dependent noise waves. By using the idea in Smoothness-focused Mesh Subdivision, noise wave problems will be solved and it will generate a smoother result. Compared with all the methods mentioned in the paper, Feature Sensitive Subdivision Framework generates a better feature preserved result.

#### ACKNOWLEDGMENT

This work was supported by National Science Foundation Grant of China 61370160, 61402120, Guangdong Province Natural Science Foundation Project 2015A030313578, 2014A030310348, Guangdong Scientific and Technological Plan Project 2015B010106005.

Corresponding author: Yongyi Gong.

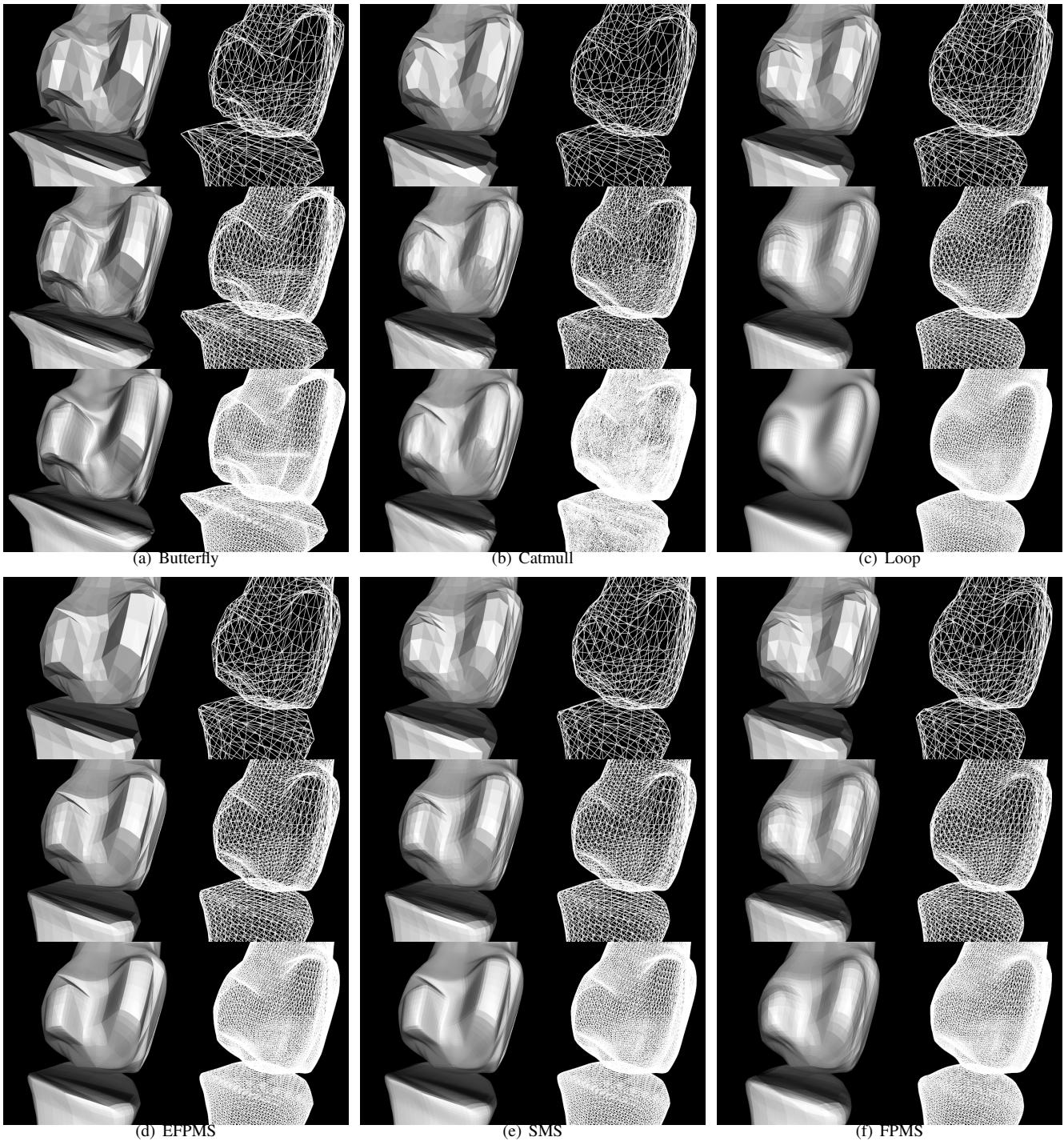


Figure 9: Subdivision Comparison on phalangeals with iteration increasing by rows. a) Butterfly Method. b) Catmull Method. c) Loop Method. d) Edge Feature Preserved Mesh Subdivision. e) Smoothness-focused Mesh Subdivision. f) Feature Preserved Mesh Subdivision.

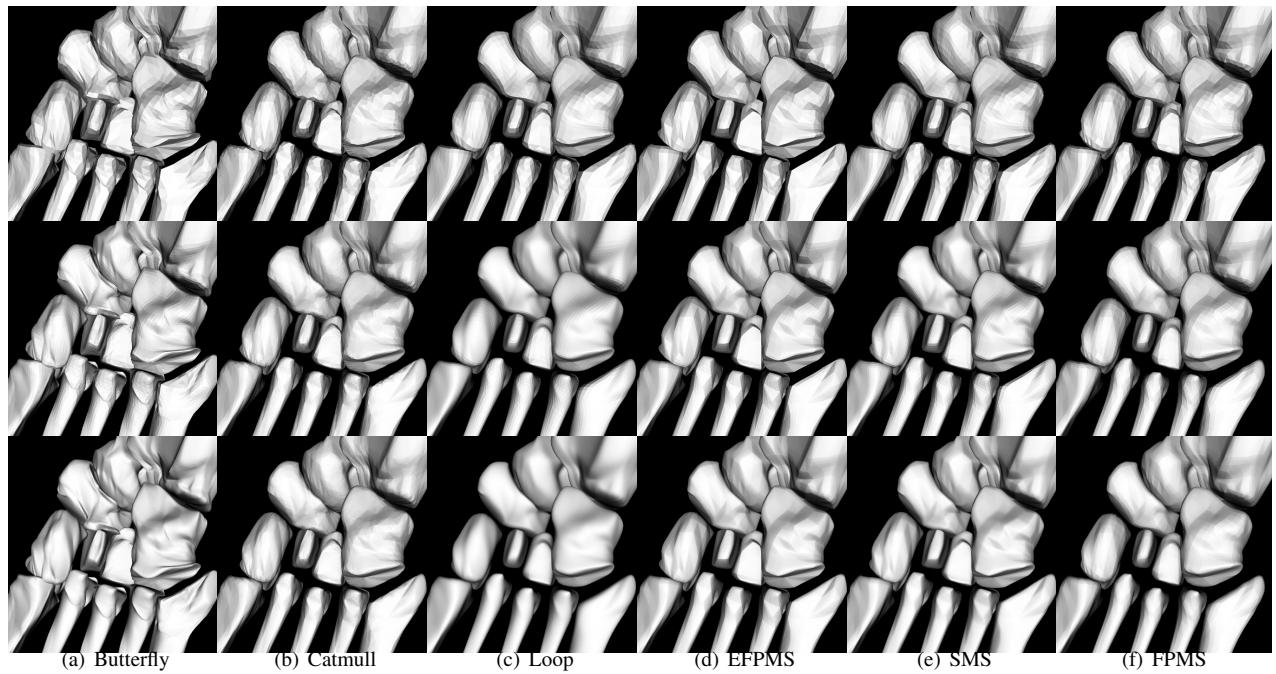


Figure 10: Subdivision Comparison on human footbone with iteration increasing by rows. a) Butterfly Method. b) Catmull Method. c) Loop Method. d) Edge Feature Preserved Mesh Subdivision. e) Smoothness-focused Mesh Subdivision. f) Feature Preserved Mesh Subdivision.

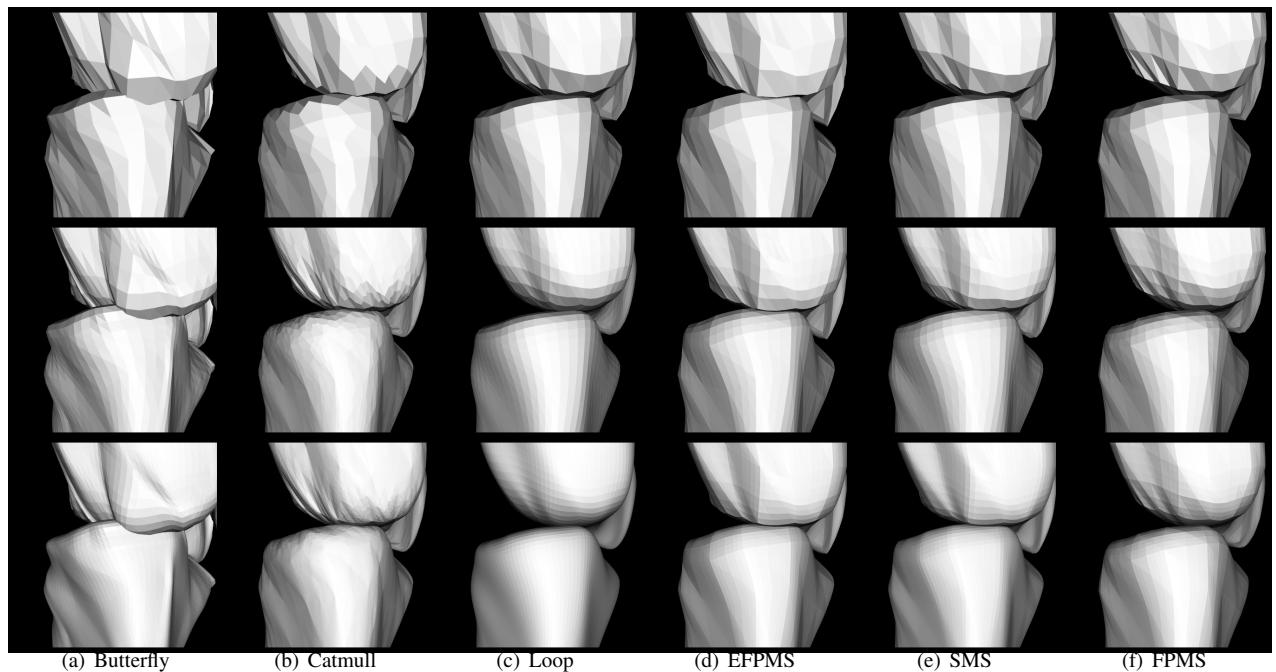


Figure 11: Subdivision Comparison on metatarsal bones and entocuneiforms with iteration increasing by rows. a) Butterfly Method. b) Catmull Method. c) Loop Method. d) Edge Feature Preserved Mesh Subdivision. e) Smoothness-focused Mesh Subdivision. f) Feature Preserved Mesh Subdivision.

## REFERENCES

- [1] J. F. Deng, L. I. Jian Xin, L. I. Lei, and B. Yan, “Fpga accelerate 3d ct reconstruction,” *Application of Electronic Technique*, 2010.
- [2] D. Koopman, J. Dalen, H. Arkies, A. B. Francken, J. Bart, S. Knollema, C. Slump, and P. Jager, “A small voxel fdg-pet/ct reconstruction improves the visual evaluation of axillary lymph nodes in patients with breast cancer,” *Journal of Nuclear Medicine*, vol. 57, no. supplement 2, pp. 1502–1502, 2016.
- [3] P. Muthusami, N. Shkumat, V. Rea, A. H. Chiu, and M. Shroff, “Ct reconstruction and mri fusion of 3d rotational angiography in the evaluation of pediatric cerebrovascular lesions.” *Neuroradiology*, vol. 59, no. 7, pp. 1–9, 2017.
- [4] M. T. Mccann, M. Nilchian, M. Stampaoni, and M. Unser, “Fast 3d reconstruction method for differential phase contrast x-ray ct,” *Optics Express*, vol. 24, no. 13, p. 14564, 2016.
- [5] X. Tao, J. Yuan, and X. Yin, “Application of 3d ct reconstruction in maxillofacial surgery,” *Acta Universitatis Medicinae Tangji*, 1999.
- [6] E. S. Lein, M. J. Hawrylycz, N. Ao, M. Ayres, A. Bensinger, A. Bernard, A. F. Boe, M. S. Boguski, K. S. Brockway, and E. J. Byrnes, “Genome-wide atlas of gene expression in the adult mouse brain.” *Nature*, vol. 445, no. 7124, pp. 168–76, 2007.
- [7] B. Bonev and G. Cavalli, “Organization and function of the 3d genome,” *Nature Reviews Genetics*, vol. 17, no. 11, p. 661, 2016.
- [8] H. Tjong, W. Li, R. Kalhor, C. Dai, S. Hao, K. Gong, Y. Zhou, H. Li, X. J. Zhou, and G. M. Le, “Population-based 3d genome structure analysis reveals driving forces in spatial genome organization.” *Proc Natl Acad Sci U S A*, vol. 113, no. 12, p. 201512577, 2016.
- [9] J. Caspermeyer, “Principles of 3d genome folding and gene expression studied across species,” *Molecular Biology & Evolution*, vol. 34, no. 6, pp. 1548–1548, 2017.
- [10] J. P. Carson, C. Thaller, and G. Eichele, “A transcriptome atlas of the mouse brain at cellular resolution,” *Current Opinion in Neurobiology*, vol. 12, no. 5, pp. 562–565, 2002.
- [11] T. Ju, J. Warren, G. Eichele, C. Thaller, W. Chiu, and J. Carson, “A geometric database for gene expression data.” *Symposium on geometry processing : [proceedings]. Symposium on Geometry Processing*, vol. 2003, p. 166, 2003.
- [12] S. Schaefer, J. Hakenberg, and J. Warren, “Smooth subdivision of tetrahedral meshes,” in *Eurographics/acm SIGGRAPH Symposium on Geometry Processing*, 2004, pp. 147–154.
- [13] J. A. Brentzen, R. Abdrashitov, and K. Singh, “Interactive shape modeling using a skeleton-mesh co-representation,” *Acm Transactions on Graphics*, vol. 33, no. 4, pp. 1–10, 2014.
- [14] I. V. Guskov, P. Schrder, and W. Sweldens, “Non-uniform relaxation procedure for multiresolution mesh processing,” 2014.
- [15] Y. Zhang and L. Ma, “A scaling method of sensitive objects based on loss constraint triangle mesh deformation,” *International Journal of Image & Graphics*, vol. 3, no. 2, 2015.
- [16] J. Tao, J. Carson, L. Liu, J. Warren, M. Bello, and I. Kakadiaris, “Subdivision meshes for organizing spatial biomedical data,” *Methods*, vol. 50, no. 2, pp. 70–76, 2010.
- [17] D. Zorin, P. Schrder, and W. Sweldens, “Interpolating subdivision for meshes with arbitrary topology,” *Acm Siggraph Computer Graphics*, vol. 30, pp. pgs. 43–53, 1996.
- [18] C. T. Loop, “Smooth subdivision surfaces based on triangles,” *Department of Mathematics the University of Utah Masters Thesis*, 1987.
- [19] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, 1978.
- [20] Q. Xie, K. Y. Sze, and Y. X. Zhou, “Drape simulation using solid-shell elements and adaptive mesh subdivision,” *Finite Elements in Analysis & Design*, vol. 106, no. C, pp. 85–102, 2015.
- [21] H. H. Liu, “An improved refinement algorithm of triangular mesh subdivision based on minimum weight theory,” *Applied Mechanics & Materials*, vol. 513–517, no. 513–517, pp. 2552–2555, 2014.
- [22] A. Amresh, G. Farin, and A. Razdan, “Adaptive subdivision schemes for triangular meshes,” pp. 319–327, 2002.
- [23] D. Rose, M. Kada, and T. Ertl, “On-the-fly adaptive subdivision terrain,” in *Vision Modeling & Visualization Conference*, 2001, pp. 87–92.
- [24] L. Kobbelt, “3-subdivision,” *Proceedings of Acm Siggraph*, vol. 18, no. 01, pp. 103–112, 2000.
- [25] S. Seeger and H. Kai, “A sub-atomic subdivision approach,” in *Vision Modeling and Visualization Conference*, 2001, pp. 77–86.
- [26] Q. Li, J. Ye, and C. Kambhamettu, “Interest point detection using imbalance oriented selection,” *Pattern Recognition*, vol. 41, no. 2, pp. 672–688, 2008.