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 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 implements linear one to four triangular mesh subdivisions illustrated in Figure 3, smoothed by proposed smooth operator and generates an edge sensitive result.

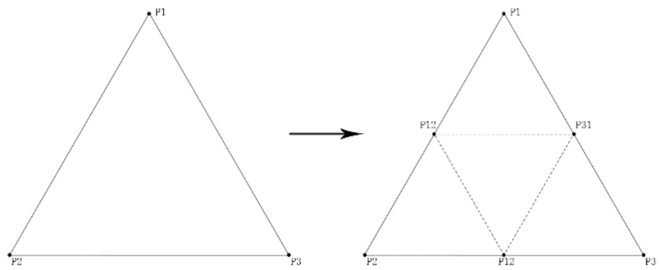


Figure 3 Linear 1 to 4 subdivision of triangles for triangular subdivision schemes.

Let's denote the original mesh as M_i. To perform linear triangular subdivision, we insert new 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 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 is then split into four sub-triangles and original mesh M_i is subdivided to M_{i+1}

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 transmuted from one-ring vertices of previous mesh M_i. For each vertex on M_{i+1}, we use a neighbor weighted centroid method for vertices as shown in Figure 4. The weight of each one-ring neighbor β is decided by the number of one-ring neighbors n.

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

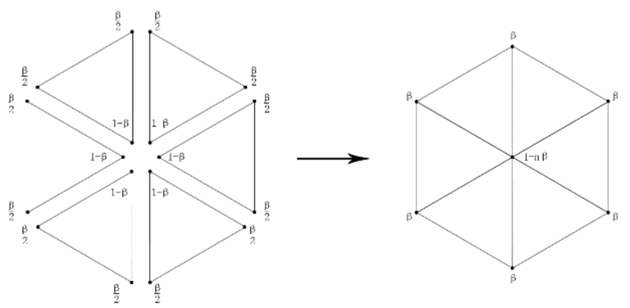
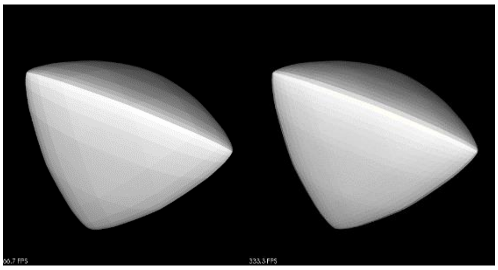


Figure 4 Neighbor weighted centroid method for triangular vertices

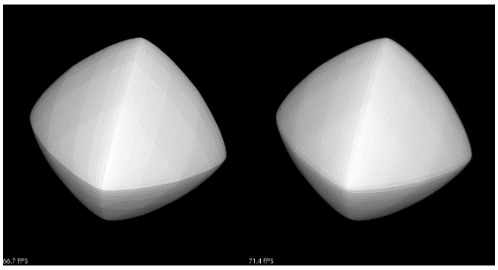
3.2. Advanced Edge Sensitive Mesh Subdivision

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 5.



a b

Figure 5 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 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.

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 neighbors as illustrated in Figure 7.

The smooth method depends on the elder vertices on M_i but newly generated vertices on M_{i+1}. The weight of each neighbor vertices 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 results 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

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 to balance the emphasis of features and smooth of subdivided mesh.

The unit of edge points generation in our provided method is a couple of triangles back to back as illustrated in Figure 7. 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.

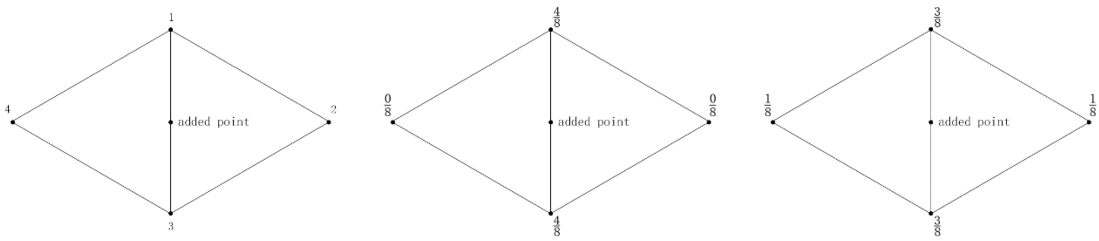


Figure 7 Edge Sensitive Subdivision edge point generator (left). Advanced Edge Sensitive Subdivision edge point generator (right)

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 $G_{AESS}(N)$

5. Experiment

6. Conclusion

7. 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