A New Progressive Mesh with Adaptive Subdivision for LOD Models

Xiaohu Ma, Jie Wu, and Xiangjun Shen

School of Computer Science and Technology,
Soochow university,Jiangsu,215006,China
xhma@suda.edu.cn

Abstract. Highly detailed models are becoming commonplace in computer graphics. Such models are often represented as triangle meshes, challenging rendering performance, transmission bandwidth, and storage capacities. This paper proposes a new scheme for constructing progressive meshes with adaptive subdivision. Besides, we introduce half-edge collapse simplification and present a new way to calculate error distance. Some experimental results show that our proposed method works well on various meshes.

1 Introduction

In order to meet the increasing needs on three-dimensional graphics, geometric models become highly detailed. However, these complex meshes are expensive to store, transmit, and render. In order to solve this problem, based on the mesh simplification algorithm, various level of detail (LOD) methods are created. Therefore people can first use the simplified meshes to replace the original meshes directly, or define a set of different levels of models, and then render them according to diverse scenes and view-points. However, if we use the general mesh simplification algorithm to construct the LODs, we have to build various simplified models, which usually requires lots of levels to assure the continuity among them. This makes effective storage and transmission impractical. Hoppe [1] proposed a progressive mesh (PM) scheme, which transform arbitrary mesh into a simplified one and stores all the necessary transform information for recovering the original mesh. This method can not only offer multiple LOD models, but also support progressive transmission in progressive mode. This paper proposes a new way to generate PM in a more appropriate way.

2 PM with Half-Edge collapse

2.1 PM model

The basic principle of Hoppe's PM is to iteratively collapse an edge from original mesh and lower the resolution gradually. At last, we may gain a coarse mesh and detailed collapse records. Based on these records, we may recover the original mesh through inserting vertices and triangles.

Figure 1 shows the edge collapse process. It shows that an edge collapse operation corresponds to a vertex split operation in an opposite action.

The problem with Hoppe's edge collapse method is that it needs to determine and create a new edge collapse point, which is time-consuming and memory-demanding. To resolve this problem, this paper adopts half-edge collapse scheme, which picks one of the two vertices on the given edge as the collapse point. Therefore, it can avoid complicated compute and massive storage requirement in creating a new collapse point.

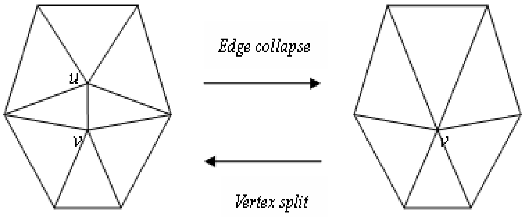

Fig. 1. Contraction of the edge (u, v) into a point.

2.2 PM generating with half-edge collapse simplification

The mesh simplification algorithm plays a critical role in PM to reduce the number of faces while keeping original model's fidelity much. During the last decade, various mesh simplification algorithm have been proposed. The representative works include Vertex decimation algorithms [2], Re-tiling mesh algorithms [3], wavelet decomposition algorithms [4] and edge collapse algorithms [5]-[8].Garland [6] proposed an efficient and effective Quadric Error Metric to weigh the cost of edge collapse, in creating the new collapse point. The problem with QEM is that it only reflects the perpendicular distance from a point to a plane of triangle, not the real distance from the point to the triangle.Our method improves QEM by first applying sum of distances from the point to the triangle to control errors and adding the cost of sharp degree [8] to gain better effects.

As shown in Figure 2,assume the perpendicular projection from point P to the plane is P'.

$$P' = aP_0 + bP_1 + cP_2; \tag{1}$$

where

$$a + b + c = 1; \tag{2}$$

Because PP' is perpendicular to the line P_0P_1 and P_0P_2 respectively, there will be:

$$(aP_0 + bP_1 + cP_2 - P) \cdot (P_1 - P_0) = 0; \tag{3}$$

$$(aP_0 + bP_1 + cP_2 - P) \cdot (P_2 - P_0) = 0; \tag{4}$$

Using the Gauss elimination method on expressions (2), (3) and (4) a,b,c can be obtained easily.

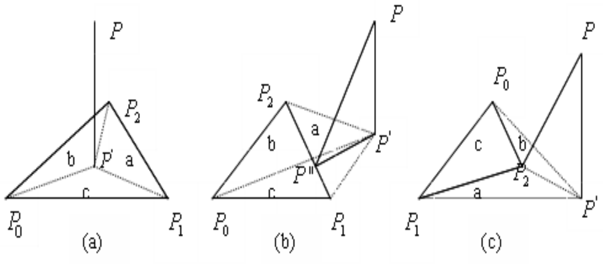

Fig. 2. Three kinds of the distances from point to triangle.(a)when P is closest to a point in the triangle (b)when P is closest to an edge in the triangle (c)when P is closest to one of three vertexes of a triangle

In order to measure the real distance from the point P to the triangle $P_0P_1P_2$, in term of the different values of a,b and c,we divide it into three kinds of instances as shown in Figure 2.
(1) If $0 \leq a, b, c \leq\ 1$,P' lies inside the triangle $P_0P_1P_2$(as shown in Fig2a),where the distance from the point P to the triangle is $|PP'|$.
(2) If one of a,b,c is smaller than 0(assume $a < 0$), P' falls outside the triangle $P_0P_1P_2$(as shown in Fig2b),where the distance from the P to the triangle is $\sqrt{|PP'|^2 + |P'P''|^2}$.
(3) If two of a,b,c are smaller than 0(assume $a < 0$, $b < 0$),P' falls outside the triangle (as shown in Fig2c),where the distance from P to the triangle is $|PP_2|$.

In order to better keeping the critical information in the original mesh, we adopt the sharp edge cost function which is ignored by Garland.

Sharp edge: Given a threshold θ and two triangles sharing a a certain edge,if the angle between the two triangles shared is more than θ,we mark the shared edge as a sharp edge.By default,we also regard the boundaries of the mesh as sharp edges.Based on this,we define the sharp degree of a vertex as the sharp edges associated with the vertex.In our cost function, we use *D(v)* to denote *v's* sharp degree. To avoid long and narrow triangles,we also consider the area cost of a vertex.

$$A(v) = \sum_{t \in N(v)} area(t) \tag{5}$$

where *N(v)* is a set of the triangles which take v as their vertex. Moreover,to prevent face-flipping phenomenon, we also check the change of the normal.If the normal is broken,we do not collapse this edge.In this paper, if the normal angle between new triangle's and former's exceeds 45 degrees, we define the normal as a broken one.In this way, our total cost function is shown as $Cost(v)$=$\alpha Q(v)$+$\beta D(v)$+$\gamma A(v)$.Here $Q(v)$ is the distance cost of half-edge collapse; α, β,γ, are set manual and the order of edge collapse is decided by the value of $Cost(v)$, from small to large.

3 PM with adaptive subdivision

As more and more 3D graphics transmit through the internet, the PM method mentioned above can implement multiple LODs better if there is enough bandwidth. However it can't fulfil the requirement when the bandwidth is limited and only a little latency time is allowed. In this paper, we adopt the subdivision algorithm to resolve the problem. The subdivision was proposed in 1970s and only received wide attention until 1990s.Various subdivision algorithms have been proposed since then.The representative researches include Catmull-Clark subdivision scheme, Loop subdivision scheme [9], Butterfly subdivision scheme and $\sqrt{3}$ subdivision scheme [10] Among them, the Loop subdivision scheme and $\sqrt{3}$ subdivision scheme become popular in recent years due to adopting triangle mesh.We can apply the loop subdivision mode in generating PM in a more effective way.If the network is not busy, we adopt the method mentioned above to get progressive mesh; if the network is busy,when the waiting time exceeds a given threshold α, we use the loop subdivision way to obtain multiple LOD models at the clients.

Generally speaking, the number of triangles increases exponentially according to a given scale in the process of subdivision. For example, in the regular Loop scheme, the number of triangles in the mesh increases three times after one subdivision. Therefore it wastes massive resources to store the information of vertex, edge and face when dealing with large-scale meshes. It is also observed that, for most surfaces, there are regions that become reasonably smooth after a few levels of subdivision, and only certain areas of the surface, where there is a high curvature change, need more subdivisions to make it smooth. Therefore it is not ideal to have a global subdivision scheme applied at every level. Adaptive algorithm offers a way to take local subdivision, by striving to get high performance with fewer faces, and controlling the rapid increase of meshes, this enable the PM express more levels in detail.

4 Experimental Results and Discussions

We have implemented the proposed method for generating PM of 3D meshes using C++ and the OpenGL graphics library.For data storage,we adopt the data structure proposed by Stan Melax [11]for easy implementation. The results obtained on many different examples have highlighted strong benefits of our approach.Figure 3 shows the progressive meshes of footbones,it shows that our method can maintain the major sharps even there are only a few faces available.Figure 4 and Figure 5 compare normal Loop subdivision and adaptive Loop subdivision which holds sharp degrees. Table 1 gives the descendent rate of faces after one subdivision while adopting adaptive algorithm. It indicates that adaptive subdivision can significantly reduce the number of faces while keeping the critical details.

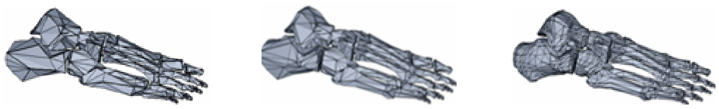

Fig. 3. The PM of footbones:the left model contains 1000 triangles,the middle one contains 2000 triangles and the right one contains 4204 triangles.

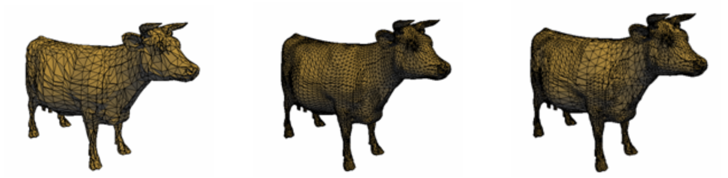

Fig. 4. The PM of cow: the left model with 4000 triangles is initial control mesh, the middle one with 16000 triangles is the result after performing one step global Loop subdivision, the right one is the result after performing one step adaptive subdivision with sharp features.

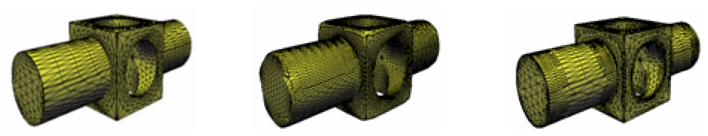

Fig. 5. The PM of mechanical part: the left model with 3808 triangles is initial control mesh, the middle one is the result after performing one step Loop subdivision, the right one is the result after performing one step adaptive subdivision with sharp features.

Table 1. The comparison between the Loop subdivision and the adaptive algorithm after performing one step subdivision respectively

Models	Loop subdivision	adaptive algorithm	Descendent rate of triangles
cow	16000 triangles	12872 triangles	19.55%
mechanical part	15232 triangles	11412 triangles	25.08%

5 Summary and future work

In this paper, we propose an efficient and effective way to generate progressive meshes based on half-edge collapse and the adaptive subdivision. Some experimental results show that our method can work well on various meshes. Main contribution of this paper is the new scheme to calculate error when collapsing the edge, and propose an adaptive subdivision method to produce PM. In the future work, we will explore to generate PM with color and texture attributes, which are important in some fields. Moreover, more systemic evaluation will be performed well.

Acknowledgements

The authors wish to acknowledge the support from the NaturalScience Foundation of Jiangsu Province under Grant 04KJB520142 and BK2007050.

References

1. Hoppe, H. Progressive meshes. In: Proceedings of the SIGGRAPH'96. 1996. 99-108.
2. Schroeder WJ, Zarge JA, Lorensen WE. Decimation of triangle meshes. Proc. of the Computer Graphics, 1992, 26(2):65-70.
3. Turk G. Re-Tiling polygonal surface. Proc. of the Computer Graphics, 1992, 26(2):55-64.
4. Lounsbery M, DeRose T, Warren J. Multiresolution analysis for surfaces of arbitrary topological type. ACM Trans. on Graphics, 1997, 16(1):34-73.
5. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. Proc. of the Computer Graphics, 1993, 27:19-26.
6. Michael Garland , Paul S. Heckbert. Simplifying surfaces with color and texture using Quadric Error Metrics. Proc. of the Computer Graphics, 1997, 31: 209-216.
7. Ma X. New methods for triangle mesh simplification based on QEM. Computer Applications, 2001, 21(12):22-24.
8. Liu X, Liu Z, Gao P, Peng X. Edge Collapse Simplification Based on Sharp Degree, Journal of Software,2005, 16(5):669-675.
9. Loop C. Smooth subdivision surfaces based on triangles [M.S.dissertation]. Department of Mathematics of University of Utah, 1987.
10. Leif Kobbelt. $\sqrt{3}$-Subdivision. In: Proceedings of Computer Graphics, Annual Conference Series, ACM SIGGRAPH, New Orleans,2000,103-112.
11. Stan Melax. A Simple, Fast, and Effective Polygon Reduction Algorithm[J].Game Developer, 1998, 11:44-49.