An Improved Quadric Error Metrics Based on Feature Matrix

Lihong Xu Weihai Chen Jingmeng Liu Tao Lv

School of Automation Science and Electrical Engineering,
Beijing University of Aeronautics and Astronautics,
Beijing, 100083, P. R. China
Email: leehomebuaa@163.com

Abstract —**The research on the base of Quadric Error Metrics (QEM) has provided some excellent eclectic methods of model approximation and time cost for mesh simplification. However, there are still some deficiencies in these methods to be settled, such as ignoring some important features and excessive simplification in some parts of the model, etc. To preserve the important geometric features, the paper has presented an improved method based on Garland's QEM by integrating a feature matrix into the approximating error with quadrics of the vertex. The new matrix based on the geometric importance and the area attribution of a vertex are considered together and used for changing the order of edge collapse in the simplification. It is shown by the experimental results that the proposed algorithm can not only be highly efficient in terms of both space and time cost, but also get highly quality approximation of the original model with the important features well preserved.**

Keyword—**Mesh Simplification, Quadric Error Metrics, Feature Value, Edge Collapse.**

I. INTRODUCTION

The transmission of a non-trivial amount of geometry data over a network has a lot of applications. First, data that is not available locally may need to be downloaded in order to be viewed and used. To avoid extremely lengthy transmission times some kind of simplification technique should be employed. Special techniques for geometry simplification have been developed for this purpose. The problem gets worse when data needs to be sent continuously, because it is being generated dynamically by the server. With simplification the primary aim is not to compress geometry in its original resolution, but to generate one or more simplified versions of a geometric model. These can be used to speed up both transmission and rendering times, for instance. As soon as several levels of resolution, i.e., detail, have been generated for a model the idea of combining all of these into a single, uniform data structure seems very worthwhile. Such multi-resolution data structures can be used to transmit geometry progressively over a network render at a level of detail that is appropriate for the current viewing distance, and a lot of other applications, like multi-resolution editing.

The goal of the mesh simplification is to reduce the mesh to a specified quantity or quality model with only an acceptable decrease in model quality, ideally while preserving as many features of the original mesh as possible. So far, there are many kinds of simplification algorithms have been proposed, including Vertex Clustering[1], Re-Tiling[2], Vertex Decimation[3,4], Wavelets[5], Edge Collapse[6, 7] and Face Constriction[8], etc. These are effective methods of simplification. Although surface simplification can reduce the number of triangles in 3D models, it is likely to destroy the contour features of the surface[9]. Edge Collapse has been identified as one of the best simplification methods in recent years. Edge collapse methods proceed in an iterative fashion in which each step involves selection of one edge for removal. Each removed edge is replaced by a vertex. The methods often seek to achieve preservation of shape by incorporating an error metric to determine which edge's removal produces the minimum change in shape.

Hoppe firstly applied edge collapse and vertex split to build a Progressive Mesh model appropriate for any 2-dimension manifold surface[10]. An optimal global energy function has been used for obtaining the collapse sequence and new vertex position. The method can always generate very accurate results. However, the expensive optimization procedure and the global error metric used in PM method require very long time to simplify large models. The method based on the Quadric Error Metric (QEM)[11] proposed by Garland is the fastest mesh simplification algorithm for all algorithms developed so far. Even though QEM is a local error control method, simplified meshes obtained by this algorithm are generally just as accurate as the ones produced by other global error control algorithms.

But the above algorithms ignore some important shape features of the original model, such as the corners and high-curvature regions, in the low-level model, and this will lead to the degeneration in the sense of sight.

In this paper we present an improved method based on Garland's QEM by integrating a feature matrix into the approximating error with quadrics of the vertex. The new matrix based on the geometric importance and the area attribution of a vertex are considered together and used for changing the order of edge collapse in the simplification. By using the improved method, the obtained model has dense meshes in the high-curvature regions and sparse meshes in the flat regions.

II. QUADRIC ERROR METRICS

A. Edge collapse

An edge collapse transformation unifies two adjacent vertices v_i and v_j into a single vertex v_n and then two adjacent faces f_2 and f_6 are removed. Except f_2 and f_6, all faces adjacent to v_i and v_j will change their shape as shown in Fig. 1. The particular sequence of edge collapse transformations determines the qualities of the approximating meshes, thus it must be chosen carefully.

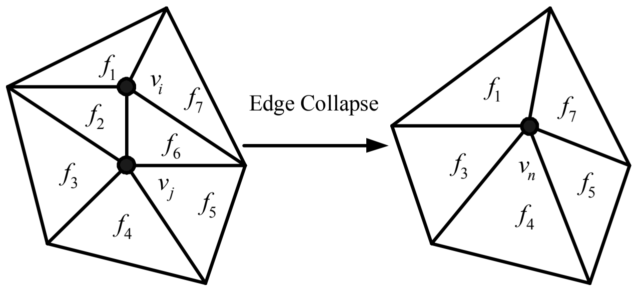

Fig. 1 Illustration of Edge Collapse

B. Position for new vertex

In the algorithm of using the operation of edge collapse, the principle of choosing the coordinates of vertex after collapsing is making the mesh shape which has been simplified similar to the original mesh as possible. However, there is no recognized measurement standard to weigh the similar extent of fore-and-after simplification mesh at present. Generally speaking, when we perform the edge collapse, which we can write $(v_i, v_j) \to v_n$, there are two main strategies of choosing the aim collapse vertex in the following.

(1) Subset choosing: $v_n = v_i$ or $v_n = v_j$, namely, we easily choose a position for new vertex v_n between v_i and v_j which are the edge's two ends depending on which one of them produces the lower collapse value. And collapsed vertices we get are the subset contained within initial model vertices.

(2) Optimizing choosing: we shouldn't choose collapse vertex limited between v_i and v_j, so much as the surface of local mesh. We should choose a position for the vertex v_n which minimizes quadric error. The corresponding quadric error matrix to v_n is:

$$Q_n = Q_i + Q_j \tag{1}$$

The quadric error of vertex v_n is a quadric equation:

$$\Delta(v_n) = v_n^T Q_n v_n \tag{2}$$

In order to find its minimum we only need to solve the equation as below:

$$\frac{\partial\Delta}{\partial x} = \frac{\partial\Delta}{\partial y} = \frac{\partial\Delta}{\partial z} = 0 \tag{3}$$

The value of x, y, z are the coordinates of new vertex v_n, this is equivalent to solving the matrix equation:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} v_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4}$$

We can get the new vertex like this:

$$v_n = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

If this matrix is not invertible, it shows there may be infinitude vertices which meet the equation. In the case we can only find the optimal vertex by the method of subset choosing.

In order to reduce the complexity of the algorithm and avoid computing the position for new vertex, the paper uses the method of subset choosing. The result of experiment shows that the method can also achieve better effect.

C. Error metrics

In the algorithm proposed by Garland it defines the error of the vertex as the sum of squared distances to its relative planes. The idiographic computing method is detailed in the following.

In three-dimensional space, one plane can be defined by equation $ax + by + cz + d = 0$ where $a^2 + b^2 + c^2 = 1$, it can also be expressed by $p^T v = 0$, $p = [a, b, c, d]^T$ is the unit normal, d is a constant. The squared distance of any vertex in the space to this plane is defined as the formula below:

$$\begin{aligned} D^2(v) &= \frac{(ax + by + cz + d)^2}{(a^2 + b^2 + c^2)^2} = (ax + by + cz + d)^2 \\ &= (p^T v)^2 = (v^T p)^2 \end{aligned} \tag{6}$$

The quadric error $\Delta(v)$ of any vertex on the mesh model is defined by the sum of squared distance to its adjacent planes:

$$\Delta(v) = \sum_{p \in planes(v)} (p^T v)^2 \tag{7}$$

In above formula, $planes(v)$ represents a set of triangles, each of which contains the vertex v, $planes(v)$ is called the set of adjacent planes about vertex v. The formula above can be rewritten as a quadric form:

$$\Delta(v)=\sum_{p\in planes(v)} v^T(pp^T)v=v^T(\sum_{p\in planes(v)} K_p)v \quad (8)$$

Here,

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (9)$$

K_p is the quadric error metrics of the plane p.

$Q=\sum_{p\in planes(v)} K_p$ is called the quadric error metrics or quadric matrix. $Q(v)=\Delta(v)$ is called the quadric error of vertex v.

When edge collapse, namely $(v_i,v_j)\rightarrow v_n$, we use the simple rule $Q_n=Q_i+Q_j$ to get the quadric error metrics for vertex v_n, the quadric error of the new vertex v_n is $\Delta(v_n)=v_n^T Q_n v_n$, and is also called the collapse error cost of the edge.

III. FEATURE MATRIX

A. Feature value

The quadric error metric algorithm proposed by Garland is a great method which achieves harmony and unification between simplification speed and simplification effect. However, there are some deficiencies in the algorithm, such as excessive simplification in some parts of the model and ignoring some important features, etc. To solve these problems, we define a feature matrix which considers the vertex curvature and the edge length together to be integrated into the quadric error of the vertex. The curvature represents the shape feature of a vertex and the length shows the influence region of the edge. By combining the vertex curvature and the edge length, the feature matrix can preferably reflect the importance of a vertex.

To get the curvature, we should calculate the normal n_v of the vertex. The simplest measure of the vertex's normal is the mean normal of its adjacent triangles. The triangle normal can be calculated by the equation as below:

$$n=\begin{pmatrix} v_2.x-v_1.x \\ v_2.y-v_1.y \\ v_2.z-v_1.z \end{pmatrix}\times\begin{pmatrix} v_3.x-v_2.x \\ v_3.y-v_2.y \\ v_3.z-v_2.z \end{pmatrix} \quad (10)$$

Here, v_1, v_2 and v_3 are three vertices of the triangle.

The vertex normal n_v can be calculated by the following equation:

$$n_v=\sum_{p\in planes(v)} n_p \quad (11)$$

Here, $planes(v)$ is the adjacent triangles of the vertex v. $n_p=n/|n|$ is the unit normal of a triangle and the vertex normal n_v commonly is united immediately.

With the vertex normal n_v, we can calculate the vertex curvature in the following equation:

$$c_v=\frac{\sum_{p\in planes(v)} \angle(n_v,n_p)}{|planes(v)|} \quad (12)$$

Here, $\angle(n_v,n_p)$ denotes the angle between n_v and n_p. $|planes(v)|$ is the number of the adjacent planes. c_v is called relative curvature, which represents the geometric importance of the vertex v. The vertex with a high value of c_v is more likely to hold salient feature of the model.

Based on the above equations, the feature value $F(v_n)$ of the edge (v_i,v_j) can be calculated as follows:

$$F(v_n)=l_{ij}^2\times\left[1+\frac{1}{2}(c_i+c_j)\right] \quad (13)$$

Where l_{ij} is the length of edge (v_i,v_j).

B. Error with feature matrix

With the feature value $F(v_n)$, the error metric of the algorithm can be calculated by the following equation:

$$\Delta'(v_n)=\Delta(v_n)+\lambda\bullet F(v_n) \quad (14)$$

Here, λ is a proportional coefficient of $F(v_n)$ which is determined by experiments.

However, Q is a 4×4 symmetrical matrix and $\lambda\bullet F(v_n)$ is a constant. For facilitating the computation, we also define $\lambda\bullet F(v_n)$ as a 4×4 matrix as follows:

$$K_f=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\lambda\bullet F(v_n)} \end{bmatrix} \quad (15)$$

The equation (14) can be rewritten as below:

$$\Delta'(v_n)=v_n^T(Q_n+K_f)v_n \quad (16)$$

In this case, the collapse cost of edges in high-curvature region and with long length can be increased significantly. With the integrated feature matrix K_f, the edges with high collapse cost will be collapsed in posterior sequence.

C. The basic steps of algorithm

We can describe the model simplification algorithm proposed by the paper as follows:

(1) Calculate the quadric error metrics for all vertices.

(2) Calculate the quadric error metrics for all edges.

(3) Calculate the feature metric for all edges.

(4) Calculate the optimal position for new vertex and the cost of collapsing the edges.

(5) Place all the edges in a heap keyed on cost with the minimum cost edge at the top.

(6) Iteratively remove and collapse the minimum cost edge from the heap, and dynamically update the information of all affected vertices.

(7) If the required model is obtained, end the program. Otherwise, turn to step (6).

IV. IMPLEMENTATION AND RESULTS

In our work, a PC with Intel Pentium 4 CPU 2.4GHz, 1G memory and Windows XP operation system has been used as the test computer. We use VC++ to validate the algorithm and OpenGL to render the model. Without loss of generality, we select three experiment models for the study, including cow model, big bunny model and car model.

A. Experimental results

Compared with the even model obtained by using the QEM algorithm with equivalent faces distributed, our experiment results which have dense meshes in the high-curvature regions and sparse meshes in the flat regions can preserves the remarkable features, as the eyes and horns of the cow shown in the Fig. 2.

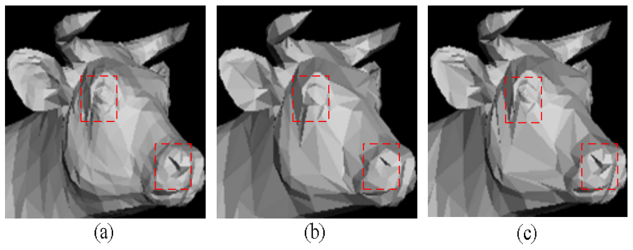

Fig. 2 Comparison of feature preserving.

(a) The prototyped cow head. (b) Preserving the features by using QEM. (c) The simplified model by using our algorithm.

Fig. 3 and Fig. 4 show the simplified results using different error metrics. Fig. 3(a) and fig. 4(a) are the original models. Fig. 3(b) and fig. 4(b) shows the simplified results using only a quadric error. In comparison, fig. 3(c)~(d) and fig. 4(c)~(d) show the ones using the quadric error metric with a feature matrix integrated. For fig. 3, when simplified to 1972 faces using quadric error metric, the model lost the sharp feature in the dashed frame as shown in fig. 3(b). But our simplified result does not lose the feature until it is simplified to 1420 faces as shown in fig. 3(c) and fig. 3(d). Similarly, compared with fig. 4(b), fig. 4(c) preserves better the particular features of the head and the neck with dense meshes. Obviously, using our method can better reflect most of the geometric features of the original model than the quadric error metrics proposed by Garland.

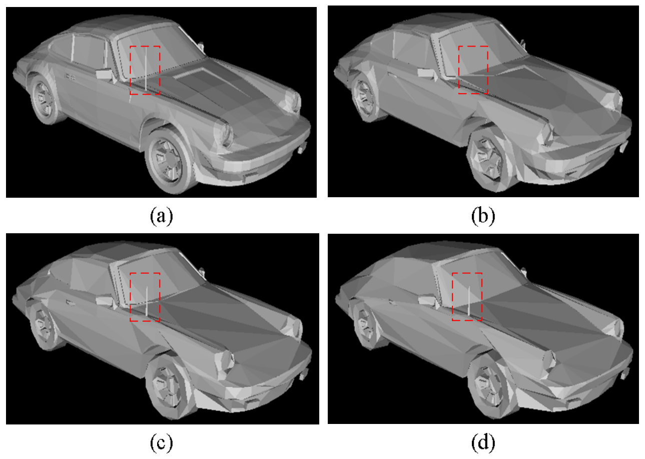

Fig. 3 Simplification of car model.

(a) The prototype with 10256 faces. (b) Feature-ignoring model with 1972 faces by using QEM. (c) Feature-preserving model with 1972 faces by using our algorithm. (d) Feature-preserving model with 1420 faces by using our algorithm.

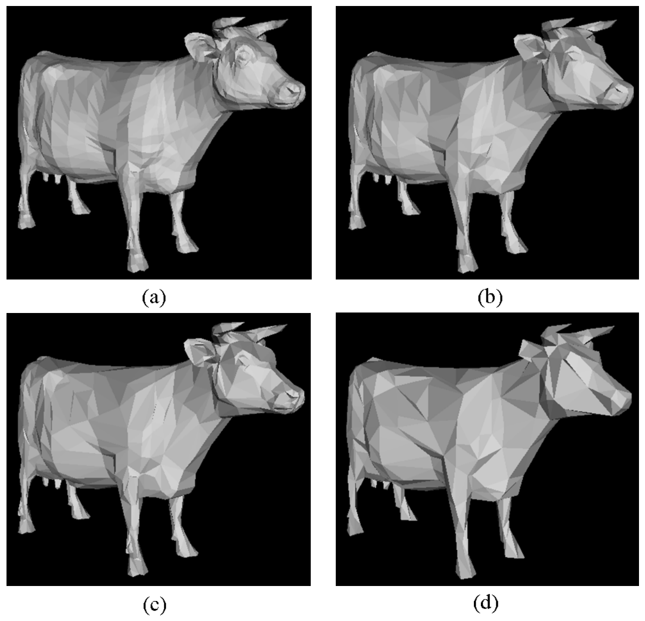

Fig. 4 Simplification of cow model.

(a) The prototype with 5804 faces. (b) The model with 1900 faces using QEM. (c) The model with 1900 faces using our algorithm. (d) The model with 700 faces using our algorithm.

B. Performance analysis

The QEM method proposed by Garland brings a very small error of the whole model. However, it also leads to the degeneration in the sense of the sight and excessive simplification in some parts of the model. In this paper, we focus on reasonably distributing the mesh and representing the salient geometric features. Commonly, most parts of the mesh model are flat but a few high-curvature regions. Accordingly, the whole error by using our method is a little more than that of QEM. However, the increased error does not influence the observation in the sense of the sight as shown in Fig. 5.

To compare the simplified errors between QEM and our method, the Cignoni's method[12] has been used for error estimation. The maximum error between two models is defined as follows:

$$E_{\max}(M_1, M_2) = \max(\max_{v \in M_1} d_v(M_2), \max_{v \in M_2} d_v(M_1)) \qquad (17)$$

Here, $d_v(M) = \min \|v - w\|$ represents the distance from a vertex v to the model M.

The average error is defined as follows:

$$E_{avg}(M_1, M_2) = \frac{1}{w_1} \int_{v \in M_1} d_v^2(M_2) + \frac{1}{w_2} \int_{v \in M_2} d_v^2(M_1) \qquad (18)$$

Here, w_1 and w_2 are respectively the areas of M_1 and M_2.

Table 1 shows the comparison of the error of the results.

Table 1 Comparison of the error of the results

Model	QEM		Our algorithm	
	Max error	Mean error	Max error	Mean error
Bunny with 5000 faces	1.212e-3	0.76e-4	0.545e-3	0.33e-4
Cow with 1500 faces	1.853e-2	0.66e-3	1.240e-2	0.45e-3
Car with 900 faces	2.874e-3	1.84e-4	2.301e-3	1.42e-4

The time complexity of Garland's quadric error metrics is $O(n \log n)$ and the consumption of the storage space is $O(n)$. Our algorithm considers the relative curvature of the vertex when computing the quadric error metrics of the vertex, and considers the factor of edge length when computing the edge's collapse cost, but holds the same complexity of the algorithm as well, which does not increase the computation time. However, the algorithm has some deficiencies at present. For example, we consider only the geometric information of the model and can not deal with the properties of the model such as color, texture and so on when simplifying the model.

V. CONCLUSION

Many simplification methods are used nowadays. They can reduce the number of triangles in models of objects. However, these methods do not account for the geometric characteristics of the object. After analyzing Garland's standard quadric error metrics and other researchers' improved algorithms, in this paper we propose a feature matrix to improve the Quadric Error Metrics, aiming at their disadvantages. The new matrix based on the geometric importance and the area attribution of a vertex are considered together and used for changing the order of edge collapse in the simplification. According to the thesis analysis and experiments' validation, the algorithm has obvious advantages of simplification effects with features well preserved which has significant meaning in practical application of mesh compression and progressive transmissions.

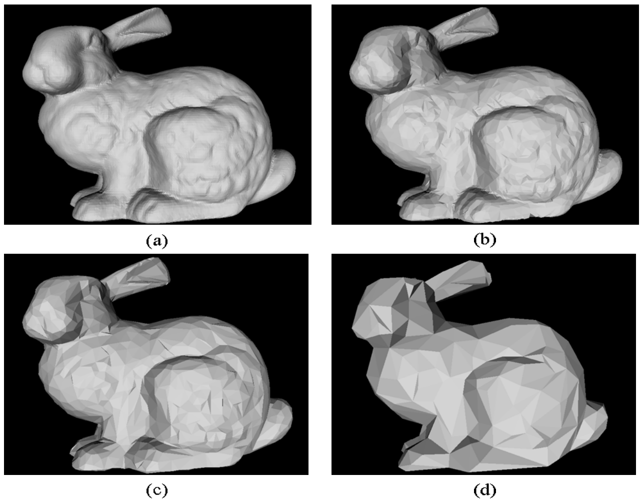

Fig. 5 Simplification of Bunny model by using our algorithm.
(a) The prototype with 69541 faces. (b) The model with 10500 faces. (c) The model with 3100 faces. (d) The model with 700 faces.

ACKNOWLEDGEMENT

This work is supported by the National High Technology Research and Development Program of China under the research project 2006AA04Z218 and Natural Science Foundation of China under the research project 60775059.

REFERENCE

[1] Low, K-L., Tan, T-S. Model simplification using vertex-clustering. Proceedings of the 1997 Symposium on Interactive 3D Graphics, ACM SIGGRAPH'97, 1997, pp. 43~50.

[2] G. Turk. Re-Tiling polygonal surfaces. Proceedings of the Computer Graphics, 1992, 26(2): pp.55~64.

[3] A.D. Kalvin, and R.H. Taylor. Surperfaces: polygonal mesh simplification with bounded error. IEEE Computer Graphics and Applications, 1996, 16(3): pp. 64~77.

[4] A. Ciampalini, P. Cignoni, C. Montani, and etc. Multiresolution decimation based on global error. The Visual Computer, 1997, 13(5): pp. 228~246.

[5] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. ACM Transactions on Graphics, 1997, 16(1): pp. 34~73.

[6] H. Hoppe. Progressive meshes. Proceedings of the SIGGRAPH'96, 1996, pp. 99~108.

[7] Jiaxin Chen, Haihe Hu. One mesh model simplification method based on shape transform of triangles. Proceedings of the 16th International Conference on Artificial Reality and Telexistence-Workshops, 2006, pp.74~79.

[8] J.H. Wu, S.M. Hu, C.L. Tai, J.G. Sun. An effective feature- preserving mesh simplification scheme based on face constriction. Proceedings of Pacific Graphics 2001, 2001, pp. 12-21.

[9] Hongtao Xu, Timothy S. Newman. An analysis of errors in feature-preserving mesh simplification based on edge contraction. Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission, 2006, pp. 134~141.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. Proceedings of the SIGGRAPH'93, 1993, pp.19~26.

[11] M. Garland, and P. S. Heckbert. Surface simplification using quadric error metrics. Proceedings of the SIGGRAPH'97, 1997, pp. 209~216.

[12] P. Cignoni, C. Rocchini, R. Scopigno. Metro: measuring error on simplified surfaces. Computer Graphics Forum, 1998, 17(2): pp.167~174