http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (2): 391-394, 2010
ISSN 1812-5638

A New Method of Mesh Simplification Algorithm Based on QEM

[1,2]Z. Tang, [1]S. Yan and [2]C. Lan
[1]School of Mechanical and Electronic Engineering,
Lanzhou University of Technology, Lanzhou, 730050, China
[2]Department of Software Engineering, Lanzhou Polytechnic College,
Lanzhou, 730050, China

Abstract: With the data increasing of 3D mesh models, in order to meet the needs of the real-time processing, the mesh models need to be simplified. The QEM is one of typical method for mesh simplification, but it only considered the distance metric, so the mesh and vertex distributing even, it is unfavorable to simplify the mesh having jags. In order to keep the important shape features and reduce visual distortion effectively at low levels of detail, in this study, we improved the QEM algorithm. First, we introduced the concept of back-cost and embed back-cost into the original garland's quadric error metric; second, we take the midpoint of contraction edge as the new vertex. The improved algorithm can not only measure distance error but also reflect geometric variations of local surface, the features of original models are well maintained. Meanwhile, new vertices are created at middle points; it is intuitional and simple than the original algorithm.

Key words: Mesh simplification, edge collapse, quadric error metric, back-cost

INTRODUCTION

In geometry modeling, the 3D shape usually is described by polygon mesh. In order to keep the realistic and hierarchy, it often takes highly complicated and detailed 3D mesh models. With the development of science and technology, it is easier and easier to generate complex mesh model, a model are made of millions even to billions cases are far from rare. This drives higher requirements of storage capacity, rendering rate and transport efficiency (Geng and Yong, 2005). In order to meet the needs of the real-time processing, the mesh models need to be simplified.

Mesh simplification has developed from static simplification into dynamic simplification, this includes: vertex cluster, region combination, re-tiling and geometric element delete algorithm is static simplification. Progressive meshes and on view-point algorithm is dynamic simplification (Guang and Jie, 2002).

Most of the existing mesh simplification algorithm is based on edge collapse. There are two critical technique of the edge collapse, one is how to select the collapse edge and another is how to select the location of the new vertex (Wei and Hao, 2009). Hoppe *et al.* (1993) took the method of energy optimization to determine the collapse edge and the location of the new vertex, the algorithm need to stand and solve the complex equation of energy optimization. Its calculation is huge and it is hard to meet the require of real-time processing, but the model result is very good. Garland and Heckbert proposed quadric error metric (QEM algorithm which solved the problem of Hoppe algorithm's huge calculation, the algorithm use quadric error metric to control the simplification process (Garland and Heckbert, 1997) and it achieve the appropriate balance between rate, fidelity and robustness, but it needs a further improvement, in this study, a method is given to improve its shortcomings.

THE THEORY OF QEM ALGORITHM

In three dimensional space, a plane's victories equation is $n^Tv+d = 0$, where, $n = [a, b, c]^T$ and $a^2+b^2+c^2 = 1$, is the unit normal vector of the plane; d is a distance constant. When completes one edge collapse, the two vertex of edge contraction to one new vertex and the plane set of new vertex is the union of two old pane's. For the distance error caused by new vertex, Garland improved the QEM algorithm, proposed the quadric methods; define it as the distance quadratic sum of the vertex to its related plane. As a result, any plane's quadric is $Q = (A, b, c) = (nn^T, dn, d^2)$, where, A is a symmetry matrix of 3×3, b is a vector of 3×1, c is a distance constant. Any distance quadratic sum of the vertex to its related plane can described in quadric as $Q(v) = D^2(v)=v^TAv+2b^Tv+c$. Because Q(v) fill the distribution law, so one plane set's quadric can obtain through sum every plane's quadric.

Corresponding Author: Zhanhong Tang, School of Mechanical and Electronic Engineering, Lanzhou University of Technology, Lan Zhou, 730050, China

Definition 1: A triangle mesh is a piecewise linear curve which connected by edge and vertex of triangles in three dimensional space and it's every edge can be included only in two triangles. Define such triangles as triangle mesh T_M, T_M can be described by (V, T) which consisted by $V = (V_1, V_2, \ldots, V_n)$ and $T = (t_1, t_2, \ldots t_m)$ (Jing and Xu-min, 2006). Vertex position is $v = [v_x, v_y, v_z, 1]^T$.

Definition 2: For any vertex V_i in T_M, called the triangle set the related triangle set of V_i if all the triangle in which include V_i and for the triangle set T_{ij}, if every triangle in which include both vertex V_i and edge e_{ij}, then called T_{ij} as related triangle set of V_i and e_{ij}.

In QEM algorithm, definite the error of vertex $v = [v_x, v_y, v_z, 1]^T$ as the distance square sum of v and its related planes(v). The error metric can described as follows:

$$\Delta(v) = \sum_{p\in planes(v)} (p^T v)^2 = \sum_{p\in planes(v)} v^T (pp^T) v = v^T \left(\sum_{planes(v)} K_p \right) v \quad (1)$$

where, p is the plane which is determinates by v (it's function is $a_x+b_y+c_z+d = 0(a^2+b^2+c^2 = 1)$) and its related triangle set, K_p is the fundamental error quadric of p.

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

$Q_v = \sum_{p\in planes(v)} K_p$ is the quadric error metric matrix of v. While initializing, since every initial vertex is the intersection points of its related triangles, the error of initial vertex is $\Delta(v) = 0$. When have the edge collapse $(v_i, v_j) \rightarrow \bar{v}$, the optimal new vertex locations and edge cost, respectively is:

$$\bar{v} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

$$\Delta(\bar{v}) = \bar{v}^T (Q_i + Q_j) \bar{v} \quad (4)$$

The quadric error metric after edge collapse is:

$$Q_{\bar{v}} = Q_i + Q_j = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (5)$$

The main advantage of this algorithm is that it is fast and its average error is low. However, because the algorithm consider only its distance metric, the mesh vertex distribute even, it is not suitable for the simplification of the model which has sharp angles.

IMPROVING THE QEM ALGORITHM

There are two critical technique of the edge collapse, one is how to select the collapse edge and another is how to select the location of the new vertex. So we improved the QEM algorithm from these two aspects.

How to calculate the cost of edge collapse: In order to embody change of the mesh surface which has the of sharp angles, we define a conception of back-cost. If m is the midpoint of collapse edge e(v,p) and p is related plane of e(v,p), then a back-cost is the angle of the m's normal and p's normal, show as Fig. 1.

We embedded the back-cost into the quadric error metric, then the quadric error metric equation based cost-back can be described as follows:

$$\begin{cases} \Delta'(v) = v^T \left(\sum_{p\in planes} cost(v,p) K_p \right) v \\ cost(v,p) = 1 - |n_m \cdot n_p| \end{cases} \quad (6)$$

where, n_m is the normal of edge e(v,p), n_p is the normal of plane p. Where:

$$n_m = \frac{\sum_{i=1}^{N} n_{m_i}}{\left\| \sum_{i=1}^{N} n_{m_i} \right\|}, \quad n_m \cdot n_p = \frac{\sum_{i=1}^{N} n_{m_i} \cdot n_p}{\left\| \sum_{i=1}^{N} n_{p_i} \right\|} \quad (7)$$

Because it is not change the accumulate nature of quadratic embedding back-cost into the quadric error metric, when going on the edge collapse $e(v_s, v_t)$, the quadratic matrix of new vertex $\bar{v}$ is $Q'_{\bar{v}} = Q'_s + Q'_t$.

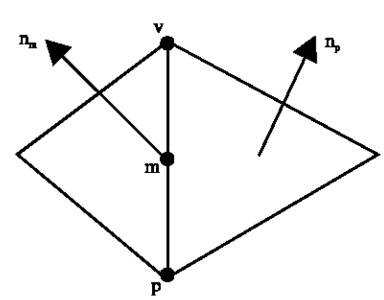

Fig. 1: The consist of back-cost

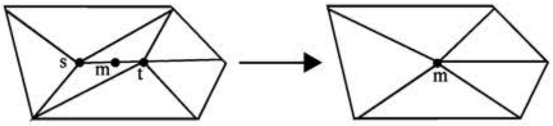

Fig. 2: The process of edge contraction based on midpoint methods

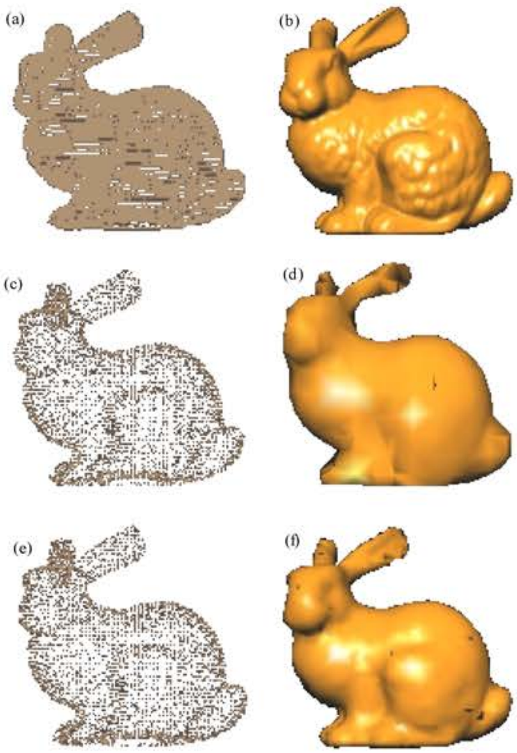

Fig. 3: The compare of original algorithm and improved algorithm. (a) The origin mesh mode, (b) the surface display of origin mesh mode, (c) the 70% mesh mode based on QEM 3.2, (d) the 70% surface display based on QEM, (e) the 70% mesh mode based on improved algorithm and (f) the 70% surface display based on improved algorithm

The location of new vertex: In this study, when we contract edge (v_s,v_t), we take the midpoint of v_s and v_t' as the new vertex , the process of edge contract is show as Fig. 2.

Experiments: According to methods mentioned above, realizing our algorithm by Visual C++6.0 and MITK (Developed by Institute of Automation Chinese Academy of Science), the result is show as Fig. 3a-f.

From Fig. 3, we can conclude that if we simply a mesh model, use our improved algorithm can get better result than the original algorithm.

DISCUSSION

After verification, using the algorithm adopted above methods, the new quadric error metric can reflect not only the distance deviation but also the extent of the model surface change. The cost of edge collapse can be restrained by the local surface geometrical change of the mesh mode. The difference of high curvature and low curvature can be increased; the high curvature's operation will be put off. On base of this, the algorithm's drawback of mesh and vertex distributing even will be improved and the model characteristic be kept better. On the same time, the selection of new vertex is intuitional and simple and easy to be realized.

CONCLUSION

Because the QEM algorithm is not consider the feature of mesh model and the complex of new vertex calculate, we proposed the mesh simplification algorithm based on midpoint. Our algorithm is an improved version of QEM, this algorithm has following characteristics: First, we introduced the concept of back-cost and embed back-cost into the original garland's quadric error metric, the conception embody the change of surface model; second, we take the midpoint of contraction edge as the new vertex, it is intuitional and simple and easy to be realized; Lastly, the new algorithm is implemented with Visual C++ and MITK and a few mesh models are simplified using this system to prove the validity of the new algorithm. But, it was showed by the experiment that it is need 1.5s to simply a model has 60'000 triangles, it is still not meet the need of real-time processing. Next, we will have research on how to use totally ordered mesh (Bouvier and Gobbetti, 2001) as the data structure and mainly focuses on how to use the data structure to reconstruction the mesh model.

REFERENCES

Bouvier, E. and E. Gobbetti, 2001. TOM: Totally ordered mesh-A multiresolution structure for time critical graphics applications. Int. J. Image Graphics, 1: 115-134.

Garland, M. and P. Heckbert, 1997. Surface simplification using quadric error metrics. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, (ACCGIT'97), ACM Press/Addison-Wesley Publishing Co., New York, USA., pp: 209-216.

Geng, P.Z. and P.M. Yong, 2005. Survey for decimation of geometric meshes. J. Jiangsu Univ., 26: 68-71.

Guang, H.H. and T. Jie, 2002. A survey on mesh simplification. J. Software, 13: 2216-2222.

Hoppe, H., T. De Rose, T. Duchamp, J. McDonald and W. Stuetzle, 1993. Mesh optimization. Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, Aug. 2-6, Anaheim, CA, USA., pp: 19-26.

Jing, C. and L. Xu-min, 2006. Multi-resolution model construction method based on half-edge collapse. Application Research of Computers, pp: 173-176.

Wei, L. and Z.D. Hao, 2009. Mesh simplification for 3d models with feature-preserving. J. Software, 20: 713-723.