Three-Dimensional Mesh Simplification by Subdivided Edge Classification

Eun-Young Chang and Yo-Sung Ho, *Member, IEEE*

Abstract-- **In this paper, we propose a new quadric error metric (QEM) to simplify 3D mesh models. The new metric is based on the subdivided edge classification to preserve object discontinuities efficiently. After we classify edges into four different types: interior, boundary, boundary incident, and complex edges, we process them according to their types and features in the 3D model. We have tested the proposed algorithm with various VRML models. Simulation results show that the proposed metric is superior to the conventional QEM in maintaining object discontinuities efficiently.**

Index Terms--**3D mesh simplification, edge classification.**

I. Introduction

In recent days, 3D models are used for various multimedia applications. Generally, 3D models are expensive to render, transmit, and store due to a large amount of information. Various mesh simplification schemes have been proposed to address this problem by reducing the size of 3D models.

One of the previous methods, the quadric error metric (QEM) [1], allows fast and accurate geometric simplification of 3D mesh models. However, it cannot capture discontinuities efficiently due to the following two reasons. First, QEM considers only a boundary edge as an exceptional case; it does not consider various characteristics of edges. Second, QEM does not consider features of the boundary edges within a model; it treats every boundary edge without discrimination.

In this paper, we propose a new quadric error metric to preserve discontinuities efficiently. This algorithm is based on the subdivided edge classification. After we classify edges into four different types: interior edge, boundary edge, boundary incident edge, and complex edge, we assign a weight to each edge according to its type and features in the 3D model.

II. Proposed Scheme

Our proposed scheme consists of three main components: iterative edge contraction, quadric error metric, and subdivided edge classification.

This work was supported in part by KOSEF through UFON, and in part by MOE through BK21.

E. Y. Chang is with ETRI (Electronics and Telecommunications Research Institute), 161 Kajong-Dong Yusong-Gu, Taejon, 305-350, Korea (telephone: +82-42-860-1677, e-mail: eychang@ etri.re.kr).

Y. S. Ho is with K-JIST (Kwangju Institute of Science and Technology), 1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea (e-mail: hoyo@kjist.ac.kr).

A. Iterative Edge Contraction

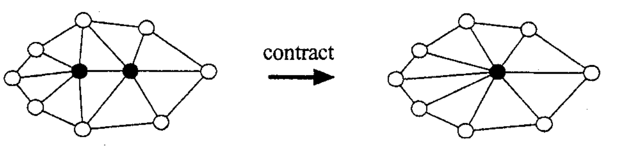
Our simplification algorithm is based on the iterative contraction operation of edges. We start with the original mesh model and remove vertices and faces from the 3D model iteratively. Each iteration involves a single atomic operation, namely, edge contraction.

Edge contraction, denoted by $(v_i, v_j) \to \bar{v}$, modifies the surface of the 3D model in the following three steps:

(1) Move the vertices v_i and v_j to the position $\bar{v}$,

(2) Replace all occurrences of v_j with v_i,

(3) Remove v_j and all faces that become degenerated.



(a) Before contraction (b) After contraction

Figure 1. Edge contraction

For the example illustrated in Fig. 1, one vertex and two faces are deleted from the mesh model.

B. Vertex Placement Policy

For the contraction operation of the edge (v_i, v_j), we need to choose the target position $\bar{v}$. In our algorithm, we simply select either v_i, v_j, or $(v_i + v_j)/2$, which produces the smallest error that is defined in Section III.

C. Subdivided Edge Classification

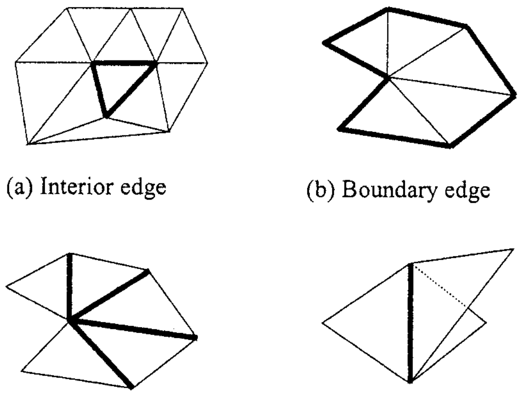
Discontinuities in the 3D model, such as creases, open boundaries, and borders between differently colored regions, are often among the most visually significant features. Therefore, preserving those features is critical for producing good quality approximations.

In this paper, we propose a subdivided edge classification algorithm to preserve such object discontinuities efficiently. Subdivided edge classification can be divided into two steps: edge classification and weight decision.

a. Edge Classification

In this step, we classify each edge into one of four different types: complex, boundary, boundary incident, and interior edge.

Fig. 2 shows four different types of edges.



(a) Interior edge (b) Boundary edge

(c) Boundary incident edge (d) Complex edge

Figure 2. Edge types

As we can see in Fig. 2, a boundary edge has exactly one incident triangle. A complex edge has three or more incident triangles. A boundary incident edge is not a boundary edge itself, but it has one or more incident boundary edges. An interior edge is neither a complex edge nor a boundary edge, and it doesn't have any incident boundary edge.

b. Weight Decision

All types of edges are candidates for deletion. However, we can assign an appropriate weight to each edge of the 3D model according to its type and features.

① Complex Edge

Since shape changes of the 3D model due to contraction of complex edges could be severe, we give a large penalty on contracting the complex edges so that the contraction operation of complex edges can be conducted later relative to other types of edges.

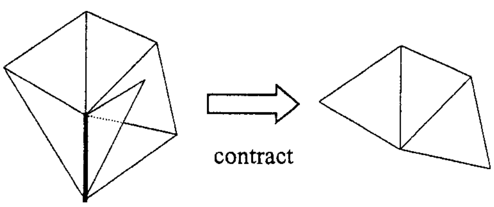


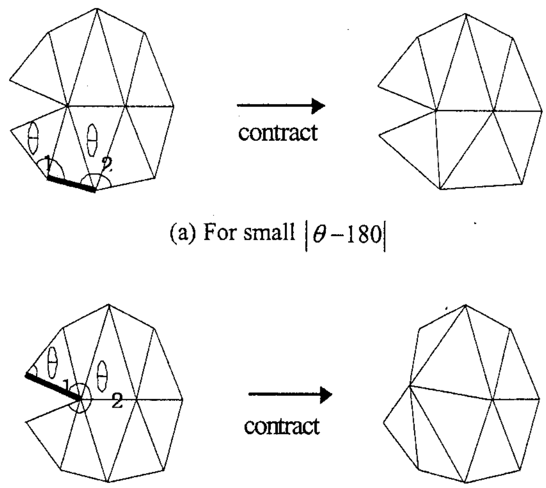
Figure 3. Contraction of the complex edge

Although we may contract the same type of edges in the 3D model, the resulting shape variations may not be the same. Thus, it is reasonable to assign different weights to edges according to their features within the 3D model. We can easily expect that the more incident triangles from the edge exist, the severer shape variation occurs. Hence, we define the *Complex_Weight* to be proportional to the number of incident triangles from the edge in the 3D model.

$$Complex\_Weight = (\#\ of\ incident\ triangles - 2)$$

② Boundary Edge

The shape variation of the 3D model becomes severe after contracting the boundary edges as well. Therefore, we must place a large penalty on contracting the boundary edges. In addition, it would be more effective if we weigh differently according to the features in the 3D model.



(a) For small $|\theta - 180|$

(b) For large $|\theta - 180|$

Figure 4. Contraction of the boundary edge

As we can see in Fig. 4, the shape variation of the 3D model after contracting the boundary edges depends on the dihedral angle, which is defined as the angle between the incident boundary edge and itself.

(1) If the value of $|\theta - 180|$ is small, the shape variation of the model is limited.

(2) If the value of $|\theta - 180|$ is large, the shape variation of the model is considerable.

Thus, we define the *Boundary_Weight* to be proportional to the value of $|\theta - 180|$.

$$Boundary\_Weight = (1 + \max\{\cos\theta_1, \cos\theta_2\})$$

③ Boundary Incident Edge

Contraction of boundary incident edges can yield substantial changes on the shape of the 3D model. Since the shape change depends on the number of incident boundary edges, we define the *Boundary_Incident_Weight* to be proportional to the number of incident boundary edges.

$$Boundary\_Incident\_Weight = (\#\ of\ incident\ boundary\ edges)/4$$

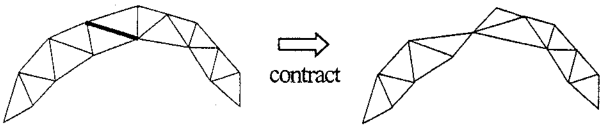


Figure 5. Contraction of the boundary incident edge

④ Interior Edge

Since shape changes of the 3D model after contracting the interior edges is milder relative to any other types of edges, no penalty on contracting the interior edges would be reasonable.

III. Error Measure for Edge Contraction

To each vertex of the 3D model, we can associate a set of triangular planes that meet at the vertex. With respect to this set, we define an error measure for the vertex as the sum of the squared distances to its planes, which is multiplied by the estimated weight.

$$\begin{aligned} Error(v) &= weight \times \sum_{p \in planes(v)} (p^T v)^2 \\ &= weight \times \sum_{p \in planes(v)} v^T (pp^T) v \\ &= weight \times v^T \left(\sum_{p \in planes(v)} (pp^T) \right) v \\ &= v^T (weight \times Q) v \end{aligned}$$

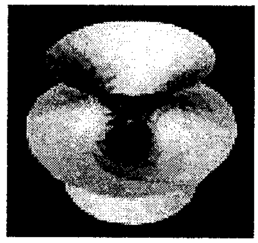
$$\begin{aligned} weight = constant\_weight \times (&1 + IsComplex \times Complex\_Weight + IsBoundary \times \\ &Boundary\_Weight + IsBoundaryIncident \\ &\times Boundary\_Incident\_Weight) \end{aligned}$$

where *IsComplex*, *IsBoundary*, and *IsBoundaryIncident* are Boolean variables. In other words, they can take TRUE(1) or FALSE(0). For example, if a given edge is a complex edge, then *IsComplex* is set to TRUE. The values of *Complex_Weight*, *Boundary_Weight*, and *Boundary_Incident_Weight* are defined in Section II and *constant_weight* is set by the user.

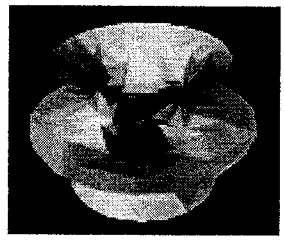
For a given contraction $(v_i, v_j) \rightarrow \bar{v}$, we must derive a new matrix $\bar{Q}$ that approximates the error at $\bar{v}$. In this paper, we use a simple addition rule, that is, $\bar{Q} = Q_i + Q_j$.

IV. Experimental Results

A. Model Simplification



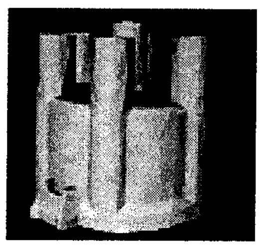
(a) Original



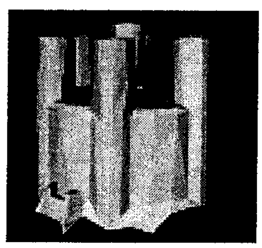
(b) QEM method (c) Proposed method

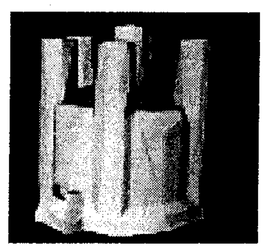
Figure 6. HYPERSHEET model

In Fig. 6(a), we show the original "HYPERSHEET" model that has 487 vertices, 917 faces and 63 open discontinuity edges. Fig. 6(b) and Fig. 6(c) show two approximations of 499 faces, i.e., having only 54% of the faces of the original model. As we can see in Fig. 6(c), the proposed scheme preserves open edge boundaries accurately.



(a) Original

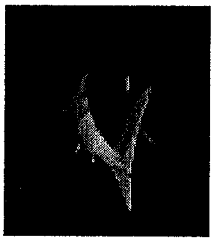




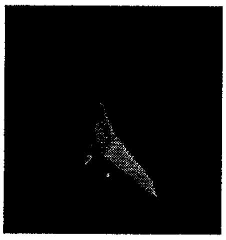
(b) QEM method (c) Proposed method

Figure 7. DISTCAP model

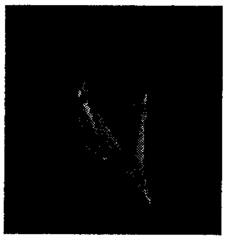
The "DISTCAP" model, shown in Fig. 7(a), has 685 vertices and 1330 faces. It has 38 open discontinuity edges. Fig. 7(b) and Fig. 7(c) show two approximations of 999 faces, i.e., 75% of the faces of the original model. By comparing Fig. 7(b) and Fig. 7(c), we can observe that the proposed scheme preserves open edge boundaries more accurately.



(a) Original



(b) QEM method



(c) Proposed method

Figure 8. SHARK model

The "SHARK" model, shown in Fig. 8(a), is a non-manifold surface that has 468 vertices and 734 faces. It has 186 open discontinuity edges and 6 complex edges. In Fig. 8(b) and Fig. 8(c), we show two approximations of 200 faces, i.e., only 28% of the faces of the original model. As we can see in the tail fin of the "SHARK" model in Fig. 8(b) and Fig. 8(c), our proposed scheme produces a more accurate approximation.

B. Distortion Measure

In order to evaluate the distortion between the original 3D polygonal model A and the reconstructed one B, we employ the Hausdorff error metric. Given two sets of points $A=\{a_1,a_2,\cdots,a_m\}$ and $B=\{b_1,b_2,\cdots,b_n\}$, the popular Hausdorff distance is defined by

$$H(A,B)=\max(h(A,B),h(B,A))$$

where

$$h(A,B)=\max_{a\in A}\left(\min_{b\in B}\|a-b\|\right)$$

Since the function $h(A,B)$ is not symmetric, it is called the directed Hausdorff distance from A to B. The Hausdorff distance $H(A,B)$ measures the degree of mismatch between two sets, as it selects the larger of the two directed distances, $h(A,B)$ and $h(B,A)$. Intuitively, if the Hausdorff distance is d, every point of A must be within the distance d from some point of B, and vice versa.

Table 1 summarizes distortion measures based on the Hausdorff distance $H(A,B)$.

Table 1. Distortion error

	HYPER-SHEET	DISTCAP	SHARK
Number of vertices (Original model)	487	685	468
Number of faces (Original model)	917	1330	734
Number of faces (Simplified model)	499 (54%)	999 (75%)	200 (27%)
Distortion measure (QEM method)	6.45	10.58	16.84
Distortion measure (Proposed method)	6.45	6.84	8.31

From Table 1, we can note that the proposed scheme is superior to the previous QEM method in terms of the Hausdorff distortion measure. Besides, we observe that the new QEM method provides improved visual quality.

V. CONCLUSIONS

We propose a new simplification scheme for 3D meshes using subdivided edge classification. We exploit various characteristics and features of edges to preserve edge discontinuities efficiently. The proposed subdivided edge classification method consists of two steps: edge classification and weight decision. After each edge is classified according to its topological characteristics, we calculate the weight of the edge according to the type and features of the edge in the 3D model. The proposed scheme has captured discontinuities efficiently and demonstrated improved performance over the conventional quadric error metric in terms of both visual quality and the Hausdorff distortion measure.

REFERENCES

[1] M. Garland and P. S. Heckbert, "Surface Simplification using Quadric Error Metrics", *SIGGRAPH*, pp.209-216, Aug. 1997.

[2] W. Schroeder and J. Zarge, "Decimation of Triangle Meshes", *SIGGRAPH*, pp.65-70, July 1992.

[3] H. Hoppe, "Progressive Meshes," *SIGGRAPH*, pp.99-108, Aug. 1996.

[4] R. Ronfard and J. Rossignac, "Full-range Approximation of Triangulated Polyhedra", 15(3), *Computer Graphics Forum*, Aug. 1996.

[5] M. Eck, T. DeRose, T. Ducham, H. Hoppe, M. Lounsbery and W. Stuetzle, "Multiresoution Analysis of Arbitrary Meshes", *SIGGRAPH*, pp.173-182, Aug. 1995

[6] E. Y. Chang, "3D Mesh Simplification using Subdivided Edge Classification," M.S. Thesis, Kwangju Institute of Science and Technology, Feb. 2001.