SHAPE-PRESERVING PROGRESSIVE CODING OF 3-D MODELS

Jeong-Hwan Ahn, Eun-Young Chang** and Yo-Sung Ho**

*Kwangju Institute of Science and Technology
1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea
Email : {jhahn,hoyo}@kjist.ac.kr

**Electronics and Telecommunications Research Institute
161 Kajong-Dong Yusong-Gu, Taejon, 305-350, Korea
Email : eychang@etri.re.kr

ABSTRACT

In this paper, we propose a shape-preserving progressive coding scheme for 3-D models, focusing on visual quality of 3-D meshes at low resolutions. In order to obtain good reconstruction quality of 3-D meshes, we improve a mesh simplification method based on the quadric error metric (QEM) by assigning different weighting factors according to topological constraints. We also develop an efficient geometric prediction algorithm that exploits neighboring information in the 3-D mesh model to achieve high compression ratios.

1. INTRODUCTION

When we transmit 3-D model data over a band-limited communication channel, incremental representation of the model is often desirable. In order to construct a progressive mesh (PM) representation [1,2], we simplify an arbitrary triangular mesh $\hat{M}$ through a sequence of edge collapse transformations, as illustrated in Figure 1, and have a simpler base model M^0.

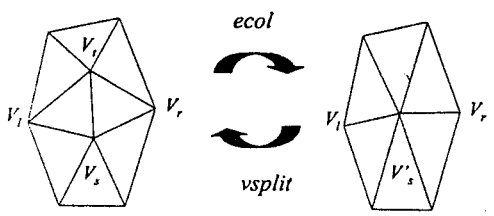


Figure 1. Edge Collapse and Vertex Split

The order of edge collapse (*ecol*) transformations can be determined by an optimization process that tries to preserve the appearance of the model as much as possible.

$$(\hat{M} = M^n) \xrightarrow{ecol_{n-1}} \dots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0$$

For each *ecol* operation, we can define a vertex split (*vsplit*) transformation to make the process reversible.

$$M^0 \xrightarrow{vsplit_0} M^1 \xrightarrow{vsplit_1} \dots \xrightarrow{vsplit_{n-1}} (M^n = \hat{M})$$

In this paper, we improve the quadric error metric (QEM) [3,4] to obtain the base model of visually good quality at low resolutions. We also devise an efficient geometric prediction algorithm based on neighbors v_i in the triangular mesh graph. Prediction errors are encoded by a context-based binary arithmetic coder [6].

2. SHAPE-PRESERVING MESH SIMPLIFICATION

Various algorithms for mesh simplification have been proposed to generate a visually pleasing sequence of progressive meshes automatically. Although the progressive mesh can provide reasonably good results, it takes very long time to simplify large models due to the expensive optimization procedure.

Even though QEM is a local error control method, simplified meshes by QEM are generally as good as those produced by other global error control algorithms. Besides, QEM is a fast algorithm. However, QEM cannot accurately preserve discontinuities, such as creases and open boundaries, which are visually significant image features.

In order to obtain good visual quality of simplified models, we classify discontinuous edges and vertices into several types according to their topological constraints. Considering those constraints, we can assign a different weight for each edge. The proposed scheme tries to avoid blind removal of edges that may result in a significant change in the visual topology of the model.

Type 1: Complex Edge

If an edge in the 3-D model is a hinge of more than two triangular faces, as shown in Figure 2, it is called as a complex edge.

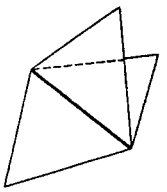


Figure 2. Complex Edge

0-7803-6725-1/01/$10.00 ©2001 IEEE

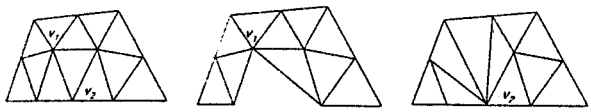
Since shape variations of the model after contracting complex edges become severe, we should give a large penalty on contraction of complex edges so that the contraction of complex edges may be conducted later relative to other types of edges. In our scheme, the weight of the complex edge is assigned proportional to the number of incident triangles from the edge in the 3-D model.

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

Type 2: Boundary Vertex

When an edge has a single vertex on the boundary, the resulting model changes according to the collapsing direction. If the vertex on the boundary is collapsed to the vertex in the interior, as shown in Figure 3(b), it destroys the overall shape of the given model. Therefore, we can collapse the interior vertex to the vertex on the boundary, as illustrated in Figure 3(c). This strategy preserves the overall shape of contracted models. Thus, we define the weight for the boundary vertex to be proportional to the number of incident edges.

$$Boundary\_Vertex = (1 + (\#\ of\ incident\ edge - 2))$$



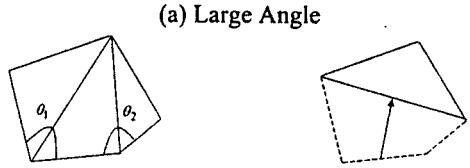
(a) (b) (c)

Figure 3. Collapsing of the Boundary Vertex

Type 3: Boundary Edge

If two vertices of an edge are on the boundary, shape variation due to the edge collapse depends on the dihedral angle at the end of the edge, as shown in Figure 4.

(a) Large Angle



(b) Small Angle

Figure 4. Criteria for the Boundary Edge

If the minimum value of two dihedral angles is small, the shape of the model is changed significantly; however, for large dihedral angles, the shape variation is minute. Therefore, we can assign the weight differently according to features of the dihedral angles.

$$Boundary\_Edge = (1 + \max(\cos\theta_1, \cos\theta_2))$$

Type 4: Interior Edge

Since shape changes of the model after contracting interior edges is milder than other types of edges, we can apply the conventional edge collapse scheme to interior edges and set the weighting factor to be the unity.

Once we assign weighting factors for different types of edges, we can define an error measure for each vertex as the sum of the squared distances to its planes, which is multiplied by the estimated weight [3,7,8].

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

3. CODING OF GEOMETRY INFORMATION

In this section, we detail the geometry coding algorithm. During the mesh simplification process, we perform the prediction operation for the next vertex position based on the removed vertex information in order to make the prediction error as small as possible.

After PM predicts the positions of the split vertex v_s and the removed vertex v_t relative to the old vertex v'_s [2], as shown in Figure 5, PM stores the prediction errors of two vertex positions, i.e., $\Delta_t = v_t - v'_s$ and $\Delta_s = v_s - v'_s$.

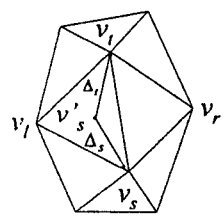


Figure 5. Geometry Prediction

When we perform the contraction $(v_t, v_s) \rightarrow v'_s$, we should determine a new contracted position v'_s carefully. For the collapsed vertex v'_s, we can select v_t, v_s, or other optimal vertex position which gives the minimum error. We can find the optimal vertex position by minimizing some quadric objective function subject to a set of linear constraints [3-5].

Figure 6 illustrates an example of edge collapse. If one endpoint of the edge is collapsed into the other endpoint, we set the index to 0 or 1. In this case, since we know the removed vertex position and two triangles, we do not need to store the collapsed vertex v'_s. Instead, we store the index and one prediction error, $\Delta_t = v_t - v'_s$.

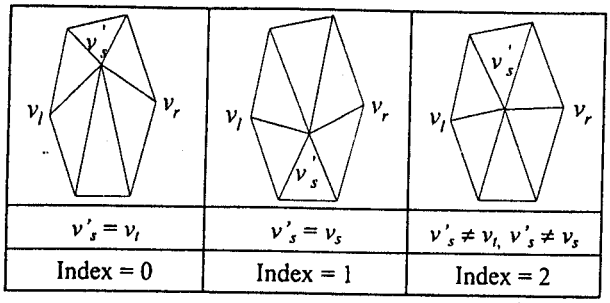
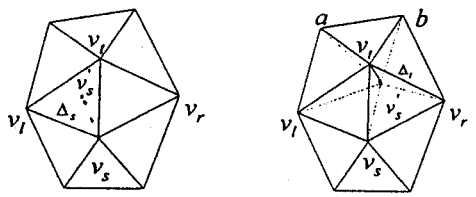


Figure 6. Prediction Rule

If the edge is collapsed into a new optimal position, instead of either endpoints of the edge, we set the index to 2 and store two position prediction errors. In this case, we calculate both Δ_s and Δ_t, as illustrated in Figure 7.



(a) Δ_s (b) Δ_t

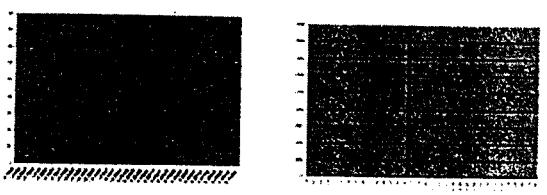
Figure 7. Calculation of Position Errors

In order to reconstruct the split vertex v_s, we define Δ_s as the difference between the split vertex v_s and the contracted vertex v'_s, as illustrated in Figure 7(a). For the removed vertex v_t, we can predict the contracted vertex v'_s as a linear combination of its immediate neighbors v_i of topological distance 1 in the triangulation graph, i.e.,

$$v'_s = \frac{\sum_{i=1}^{k} v_i}{k}.$$

In Figure 7(b), the contracted vertex v'_s is approximated by the average value, $(a+b+v_r+v_l+v_s)/5$. Then, we can obtain the prediction error, $\Delta_t = v_t - \hat{v}_s$.

Experimental results for the COW model are demonstrated in Figure 8. For the given probability distribution, we apply entropy coding to encode the prediction errors. In this work, we first quantize floating-point coordinate values of the prediction error with m bits. The residuals are encoded by the QM coder [6].



(a) Δv_t (b) Δv_s

Figure 8. Distribution of Prediction Errors (COW Model)

4. DISTORTION MEASURE

In order to evaluate the distortion between the original 3-D polygonal model A and the reconstructed one B, we employ the Hausdorff error metric [7]. Given two sets of points $A=\{a_1,a_2,...,a_m\}$ and $B=\{b_1,b_2,...,b_n\}$, the popular Hausdorff distance is defined by

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

where

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

Since the function $h(A,B)$ is not symmetric, it is called as 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.

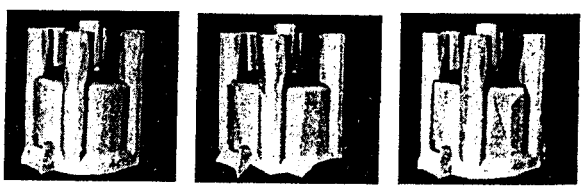
5. EXPERIMENTAL RESULTS

In this section, we analyze experimental results of our mesh simplification and geometry coding algorithms with several VRML models.

5.1. Simplification Scheme

5.1.1. Manifold Model

The DISTCAP model, shown in Figure 9(a), has 685 vertices, 1330 faces and 38 open discontinuous edges. Figure 9(b) and Figure 9(c) show two approximations of 999 faces, i.e., 75% of the original model, by the QEM and proposed simplification schemes, respectively. From Figure 9(b) and Figure 9(c), we can see that the proposed scheme preserves open edge boundaries more accurately.

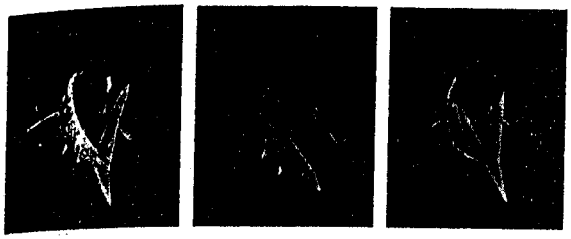


(a) Original (b) QEM (c) Proposed

Figure 9. DISTCAP Model

5.1.2. Non-manifold Model

The SHARK model, shown in Figure 10(a), is a non-manifold surface that has 468 vertices and 734 faces. It has 186 open discontinuous edges and 6 complex edges. In Figure 10(b) and Figure 10(c), we show two approximations of 200 faces, i.e., only 28% of the original model. As we can see in the tail fin of the SHARK model in Figure 10(b) and Figure 10(c), our proposed scheme produces a more accurate approximation [7, 8].



(a) Original (b) QEM (c) Proposed

Figure 10. SHARK Model

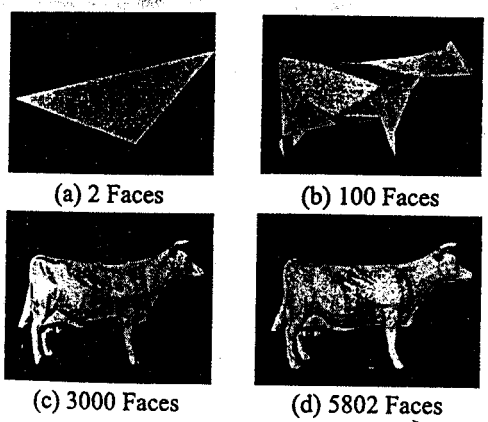
Table 1 summarizes distortion measures based on the Hausdorff distance *H(A,B)*. From Table 1, we can note that the proposed scheme is superior to the QEM method in terms of the Hausdorff distortion measure. In addition, we observe that the proposed scheme provides reconstructed 3-D models of improved visual quality.

Table 1. Distortion Measures

	DISTCAP	SHARK
QEM	10.58	16.84
Proposed Method	6.84	8.31

5.2. Geometry Coding

The COW model has 5802 faces and 2904 vertices. Its rendered images at four different resolutions and their corresponding face numbers are shown in Figure 11.



(a) 2 Faces (b) 100 Faces

(c) 3000 Faces (d) 5802 Faces

Figure 11. Progressive Reconstruction

For geometry encoding, we use 13 bits to represent each coordinate value of the vertex in the COW model. Prediction errors are then encoded by the QM coder.

Figure 12 compares performance of the progressive mesh (PM) and the proposed geometry coding schemes with approximation models of different numbers of faces. In Figure 12, the horizontal axis shows the number of removed faces and the vertical axis represents the size of the compressed file. From Figure 12, we can see that the proposed scheme is more efficient than the progressive mesh scheme in representing the geometry data.

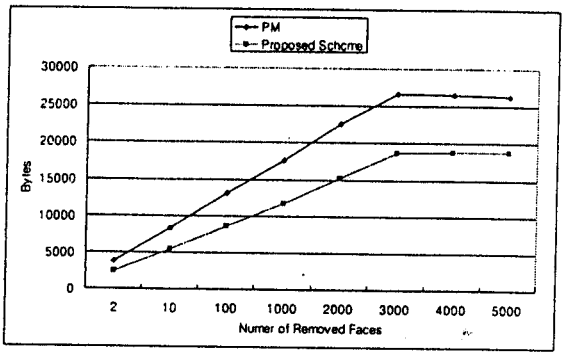


Figure 12. Performance of Geometry Encoding

6. CONCLUSIONS

In this paper, we propose a shape-preserving progressive mesh coding scheme for 3-D model representation. In order to generate visually pleasing approximations of the 3-D model, we classify discontinuous edges and vertices into several types based on their topological constraints. We also present an efficient progressive coding algorithm for geometry information. Experimental results demonstrate that the proposed scheme outperforms the previous QEM scheme, both in reconstructed visual quality and in the Hausdorff distortion measure.

ACKNOWLEDGMENT

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

REFERENCES

[1] H. Hoppe, "Progressive Meshes," *SIGGRAPH 96*, pp.99-108, 1996.

[2] H. Hoppe, "Efficient Implementation of Progressive Meshes," *SIGGRAPH 98*, pp.27-36, 1998.

[3] M. Garland and P.S. Heckbert, "Surface Simplification using Quadric Error Metrices," *SIGGRAPH 97*, pp.209-216, 1997.

[4] M. Garland and P.S. Heckbert, "Simplifying Surfaces with Color and Texture using Quadric Error Metrics," *Visualization 98 Conference Proceedings*, 1998.

[5] P. Lindstrom and G. Turk, "Evaluation of Memoryless Simplification," *IEEE Trans. on Visualiz. and Computer Graphics*, Vol. 5, No.2, pp.98-115, 1999.

[6] W.B. Pennebaker and J.L. Mitchel, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

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

[8] E.Y. Chang and Y.S. Ho, "Three-Dimensional Mesh Simplification by Subdivided Edge Classification," *IEEE Region 10 Annual Conference*, Aug. 2001.