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 Yuseong-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](image)

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.

\[
(\hat{M} = M^n) \xrightarrow{\text{ecol}_1} \ldots \xrightarrow{\text{ecol}_i} M^1 \xrightarrow{\text{ecol}_i} M^0
\]

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

\[
M^0 \xrightarrow{\text{vsplit}_1} M^1 \xrightarrow{\text{vsplit}_i} \ldots \xrightarrow{\text{vsplit}_i} (M^n = \hat{M})
\]

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](image)
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.

\[
\text{Complex Weight} = (\# \text{ of incident triangles})
\]

**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.

\[
\text{Boundary Weight} = (1 + (\# \text{ of incident edge} - 2))
\]

![Figure 3. Collapsing of the Boundary Vertex](image)

**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.

![Figure 4. Criteria for the Boundary Edge](image)

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.

\[
\text{Boundary Edge} = (1 + \text{max}(\cos \theta_1, \cos \theta_2))
\]

**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].

\[
\text{Error}(v) = \text{weight} \times \sum_{p \in \text{plane}(v)} (p^T v)^2
\]

\[
= \text{weight} \times \sum_{p \in \text{plane}(v)} v^T (pp^T) v
\]

\[
= \text{weight} v^T \left( \sum_{p \in \text{plane}(v)} (pp^T) \right) v
\]

\[
= v^T (\text{weight} \times Q) v
\]

### 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 \(v_i\) and the removed vertex \(v_{r} \), relative to the old vertex \(v'_i\) [2], as shown in Figure 5, PM stores the prediction errors of two vertex positions, i.e., \(\Delta_i = v_i - v'_i\) and \(\Delta_s = v_s - v'_s\).

![Figure 5. Geometry Prediction](image)

When we perform the contraction \((v_i, v_s) \rightarrow v'_i\), we should determine a new contracted position \(v'_i\) carefully. For the collapsed vertex \(v'_i\), we can select \(v_i, v_s\), 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 \(v'_i\). Instead, we store the index and one prediction error, \(\Delta_i = v_i - v'_i\).
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

$$h(A, B) = \max_{a \in A} \min_{b \in B} |a - b|^2.$$ 

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.

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].
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.

Table 1. Distortion Measures

<table>
<thead>
<tr>
<th></th>
<th>DISTCAPP</th>
<th>SHARK</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEM</td>
<td>10.58</td>
<td>16.84</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>6.84</td>
<td>8.31</td>
</tr>
</tbody>
</table>

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