LETTER

Predictive Geometry Compression of 3-D Mesh Models Using a Joint Prediction

Jeong-Hwan AHN†, ***Student Member and*** **Yo-Sung HO**†, ***Regular Member***

SUMMARY In this letter, we address geometry coding of 3-D mesh models. Using a joint prediction, the encoder predicts vertex positions in the layer traversal order. After we apply the joint prediction algorithm to eliminate redundancy among vertex positions using both position and angle values of neighboring triangles, we encode those prediction errors using a uniform quantizer and an entropy coder. The proposed scheme demonstrates improved coding efficiency for various VRML test data.
key words: *geometry compression, 3-D mesh, VRML, joint prediction, vertex data*

1. Introduction

In general, 3-D models are structured by polygonal meshes, which are defined by connectivity, geometry, and attribute data. While the connectivity data describe the connectivity information among vertices and characterize the topology of the model, the geometry data specify the overall shape of the 3-D mesh model. The attribute data specify information of each surface, such as colors, surface normals, and texture coordinates [1].

The connectivity coder produces an efficient representation of the association between each mesh and its sustaining vertices. The attribute coder is for lossy or lossless compression of color, normal vectors and texture coordinates data. The geometry coder is for lossy or lossless compression of vertex positions. Among these data, the geometry data specified by three floating-point numbers take the largest part of the 3-D model representation. In this letter, we propose a geometry compression scheme.

Most of the existing schemes for 3-D mesh model coding include a predictive coding structure to remove redundancies among vertex positions based on observations of the previously coded vertex positions. For example, Deering has conducted on 3-D mesh compression based on generalized triangle strips [2]. Linear quantization is applied to properties along the generalized triangle strip. A bounding box is defined to convert from floating-point representation to fixed-point representation. The fixed point numbers are then differentially coded and Huffman coding is further applied.

Taubin and Rossignac introduced a topological surgery scheme, where the current vertex is linearly predicted along the vertex spanning tree [3]. Although they have introduced coding techniques, their emphasis was not so much on coding efficiency.

Touma and Gotsman proposed an encoding scheme with implicit traversal of the mesh based on the degree of each vertex [4]. The geometry of the mesh is differentially coded with respect to the predicted value by the so-called parallelogram rule. The difference between the predicted and the actual values is then Huffman coded. Bossen has applied this parallelogram prediction rule to the topological surgery and increased coding efficiency by the QM coder [5]. Later, this geometry coding scheme has been adopted as the MPEG-4 SNHC standard [6].

In this letter, we propose a new coding scheme based on the DPCM structure to exploit the strong correlation of both coordinate and angle values among neighboring vertices in 3-D mesh models. After we estimate the current vertex position based on observations of the previously coded vertex positions in the layer traversal order, the difference between the original and the predicted vertex coordinate values is encoded by a uniform quantizer and an entropy coder.

2. Geometry Coding of 3-D Models

Figure 1 shows a block diagram of the proposed geometry encoder. Conventional geometry coding schemes [2]–[6] quantize vertex positions using a bounding box before coding them in a lossless manner. However, in

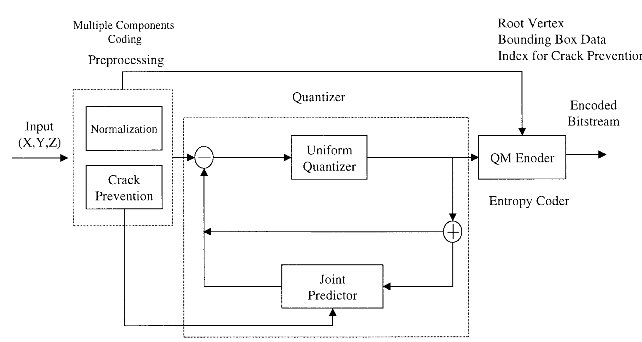


Fig. 1 Block diagram of a geometry encoder.

Manuscript received March 7, 2002.
Manuscript revised May 31, 2002.
†The authors are with the Visual Communications Laboratory, Department of Information and Communications, Kwang-Ju Institute of Science and Technology (K-JIST), 1 Oryong Dong, Buk Gu, Kwang-ju 500-712, Republic of Korea.

the proposed scheme, we first predict the vertex positions and then quantize them differentially [8], [9]. Our geometry encoder consists of four stages: preprocessing, prediction, quantization, and entropy coding.

2.1 Preprocessing

2.1.1 Multiple Component Coding

In general, 3-D mesh models are composed of multiple components. A vertex of each component is selected as a root vertex. The root vertex plays a pivotal role as the anchor point for the entire component. Since the root vertex of each component has no preceding vertices, this root vertex cannot be coded differentially as the other vertices. However, independent coding of the root vertex position with its own floating-point numbers is not efficient. Therefore, the root vertex position is predicted by the last vertex position of previously coded components.

Sometimes, several components of a 3-D mesh object are created independently and put together by sharing some vertices. More specifically, when a 3-D mesh model is generated using a typical authoring tool, some parts are often duplicated through a cut-and-paste process, while other parts are grabbed out of readily usable object libraries. Whenever such independently created parts are placed to have shared vertices, a small gap may occur between boundaries of two adjoining components due to a limited precision of the authoring tool. This anomaly is called the crack problem [7], [9]. In order to solve this problem, we design a preprocessor to identify shared vertices that will be used for the crack prevention and to enforce them to have the same value. In our scheme, a list of shared vertices is delivered to the decoder. The side information enables the encoder to skip the encoding of duplicate descriptions of shared vertices. Thus, for 3-D mesh models having many shared vertices, this approach also leads to a significant reduction in the overall bit rate.

2.1.2 Normalization

In the normalization stage, we calculate the tightest bounding box containing the 3-D mesh model and then normalize the vertex coordinates by Eq.(1) so that they lie within the interval $[0,1] \times [0,1] \times [0,1]$.

$$\acute{v} = \frac{v - \min(v)}{\max(v) - \min(v)} \tag{1}$$

The normalization is needed to limit the dynamic range of prediction errors for vertex coordinates, since excessively large residuals or prediction errors can lead to visually unacceptable geometric distortion, as shown in Fig. 2.

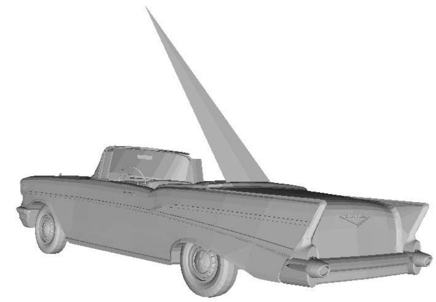


Fig. 2 Visually unacceptable geometric distortion.

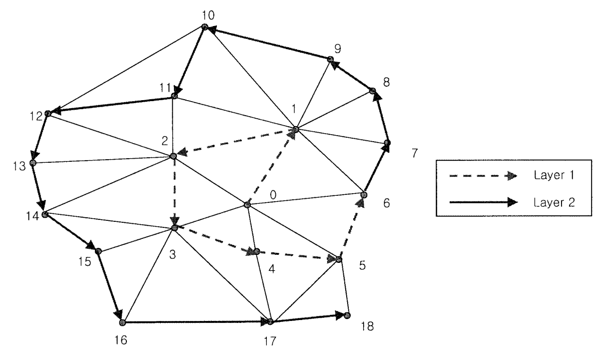


Fig. 3 Layer traversing.

2.1.3 Layer Traversing

The main reason to order vertices is to optimize the prediction operation for the geometry coding. In the proposed scheme, we set vertex ordering in a coherent fashion. In this way, each vertex can be predicted accurately by its neighboring vertices. Therefore, it produces residual errors of a good-shaped distribution and improves coding efficiency significantly.

The mesh traversal is carried out as follows. Once we begin from a randomly selected initial vertex, we visit the next vertex of topological distance 1 in the triangular graph in the counterclockwise direction, where the center vertex is called a pivot vertex. After all vertices of topological distance 1 around the pivot vertex are visited, we find vertices of topological distance 2 in the triangular graph. During the traversal, a queue Q of vertices is constructed to record the traversal order of the vertices. If the model consists of only one connected component, all vertices are visited at the end of traversal. Figure 3 shows an example of vertex ordering on a mesh, where labels of the vertices indicate the order by which they are traversed.

The prediction operation of the vertices is strictly a causal process. A vertex can be predicted only by the preceding vertices. In our proposed algorithm, a vertex position is predicted by those of its neighboring vertices.

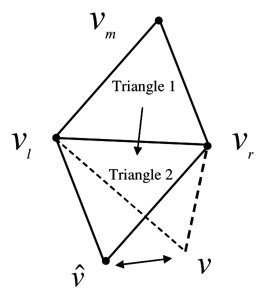


Fig. 4 Parallelogram prediction.

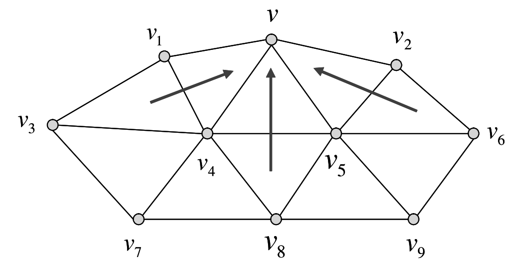


Fig. 5 Average parallelogram prediction.

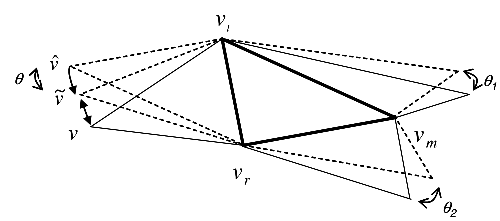


Fig. 6 Angle prediction.

2.2 Prediction

2.2.1 Parallelogram Prediction

In order to take a differential coding approach, we obtain a prediction value $\hat{v}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$ of each vertex point $v_n = (x_n, y_n, z_n)$ based on the parallelogram prediction [4], as illustrated in Fig. 4. The parallelogram prediction is well adapted for the triangle strip along the layer traversing. When we pass from Triangle 1 to Triangle 2, the vertices v_l, v_r, and v_m are already decoded. The opposite vertex v from the common edge (v_l, v_r) is predicted as $v_l + v_r - v_m$. Thus, the predicted $\hat{v}$, together with its three ancestors, forms a parallelogram and belongs to the same plane.

However, the parallelogram prediction scheme only uses three previously traversed vertices in one adjacent triangle so that the predicted vertex can be located at the biased position from neighboring vertices. In addition, all vertices are assumed in same plane. Thus, the vertex on a curved face may not be predicted accurately.

2.2.2 Joint Prediction

In the proposed scheme, we predict the position of a vertex v using geometrically nearest vertices which is already traversed by the layer traversing. In order to exploit local correlation between neighboring vertices, we employ every traversed triangle to approximate the predicted vertex $\hat{v}$. If three triangles are already traversed for vertex v, the prediction is applied for each triangle, as illustrated in Fig. 5.

Depending on how many preceding vertices exist, the prediction is conducted in four different ways:

$$\hat{v} = \begin{cases} v_l & \text{if } v_l \text{ is available} \\ v_r & \text{if } v_r \text{ is available} \\ 0.5v_l + 0.5v_r & \text{if } v_r \text{ and } v_l \text{ are available} \\ v_l + v_r - v_m & \text{if } v_r, v_l, \text{ and } v_m \text{ are available} \end{cases} \quad (2)$$

The parallelogram prediction assumes that all vertices are co-planar. However, the triangle is generally creased along the shared edge. If we estimate the dihedral angle θ between two triangles, we can obtain a more accurate vertex position.

We approximate the dihedral angle θ as the average of the other dihedral angles θ_1 and θ_2 between two sets of adjacent triangles, as shown in Fig. 6.

After estimating the dihedral angle θ, we project the predicted vertex $\hat{v}$ into $\tilde{v}$ by Eq.(3), where $\hat{v}$ is calculated by Eq.(2).

$$\tilde{v} = \|v\| \frac{\hat{v}\cos\theta}{\|\hat{v}\cos\theta\|} \quad (3)$$

where

$$\theta = \begin{cases} \theta_1 & \text{if } \theta_1 \text{ is available} \\ \theta_2 & \text{if } \theta_2 \text{ is available} \\ \frac{(\theta_1+\theta_2)}{2} & \text{if } \theta_1 \text{ and } \theta_2 \text{ are available} \end{cases}$$

As a result, the estimated vertex $\acute{v}$ is defined as

$$\acute{v} = \frac{1}{N}\sum_{k=1}^{N} \tilde{v}_k = \frac{1}{N}\sum_{k=1}^{N} \|v\| \frac{\hat{v}\cos\theta}{\|\hat{v}\cos\theta\|} \quad (4)$$

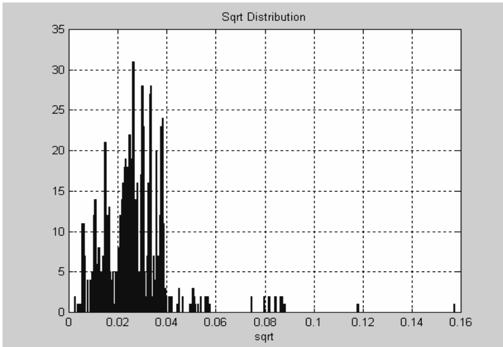
where N is number of the preceding triangles for the given vertex v.

Distribution of prediction errors Δv for the model EIGHT is demonstrated in Fig. 7. The prediction errors $\Delta v = v - \acute{v}$ are defined as $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, where Δx, Δy, and Δz are prediction errors in each direction, respectively. Figure 7(a) shows the result of the parallelogram prediction and Fig. 7(b) shows the result of the proposed prediction scheme.

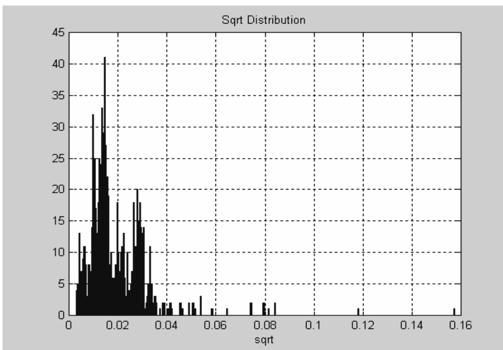
From Fig. 7, we can observe that the proposed scheme has more concentrated prediction errors near zero than the parallelogram predictor.

2.3 Quantization and Entropy Coding

After we estimate the current vertex position based on



(a) Parallelogram prediction.



(b) Joint prediction.

Fig. 7 Distribution of prediction error for model EIGHT.

Table 1 Test models.

Model	nV	nF	nC
HORSE	11,135	22,258	3
SKULL	10,952	22,10	1
BEETHOVEN	2,845	2,812	20
CROCODILE	17,332	21,590	65
CAM-SHAFT	54,898	52,500	209
57CHEVY	18,472	15,369	585

observations of the previously coded vertex positions, the residual coordinate values are uniformly quantized [9]. Since we normalize the vertex position in the preprocessing stage, the dynamic range of prediction errors is limited to $[-1.0, 1.0]$. Thus, we can design a uniform deadzone quantizer with a tight bound of prediction errors. The quantizer index is encoded by a QM entropy coder.

3. Experimental Results

We have tested performance of the proposed 3-D model coding algorithm with some VRML models. Properties of selected test models are summarized in Table 1.

In image coding, the root mean squared error (RMSE) or the peak-signal-to-noise ratio (PSNR) is popularly employed as an objective distortion measure. Although, these error metrics are not always consistent with perceived signal quality, they allow us to estimate subjective quality to some extent. However, in 3-D model coding, because excessively large quantization errors in just a few vertices may lead to a catastrophic deformation of the shape of the 3-D model, RMSE or PSNR cannot represent the overall geometry distortion appropriately. Therefore, the Hausdorff distance, which is a max-min distance measure between two sets of points, is more meaningful in 3-D geometry compression. In order to evaluate 3-D model coding schemes, we adopt the Hausdorff distance that is defined as follows.

Let A and B denote the original 3-D polygonal model and the reconstructed one, respectively. Given two sets of points $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, the Hausdorff distance is defined by

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{5}$$

where $h(A, B) = \max(\min \|a - b\|)$.

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 in A must be within the distance d from some point in B, and vice versa.

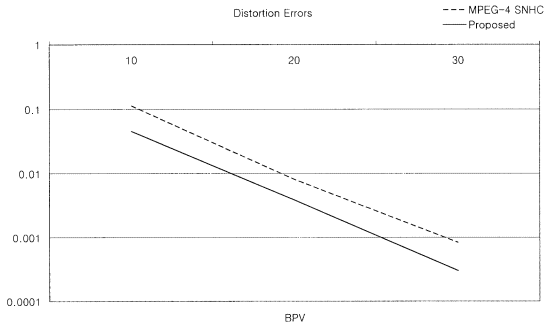
In Fig. 8, we compare our proposed scheme with a MPEG-4 standard scheme [6] for different 3-D models. The MPEG scheme is based on topological surgery [3], [5]. In Fig. 8, the horizontal axis shows the bits per vertex (bpv) and the vertical axis represents distortion error in the logarithm scale. As we can see, the proposed scheme outperforms the MPEG-4 standard.

4. Conclusions

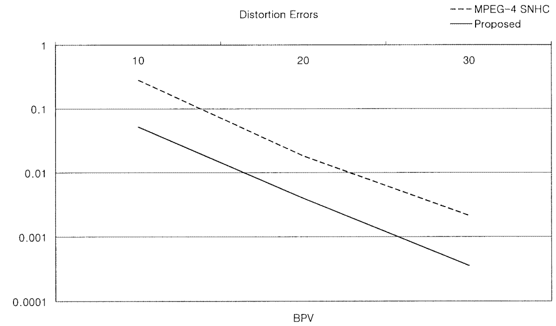
In this letter, we propose a new coding scheme for 3-D geometry information. The proposed scheme exploits the geometrical correlation and the curvature between neighboring vertices. Thus, we can obtain higher prediction accuracy than the MPEG-4 SNHC scheme. It was demonstrated that the proposed scheme outperforms the MPEG-4 SNHC scheme for the several test models. However, the overall computational requirement of the proposed scheme is slightly higher than the MPEG-4 SNHC scheme because we examine all the adjoining vertices and calculate their dihedral angles.

Acknowledgement

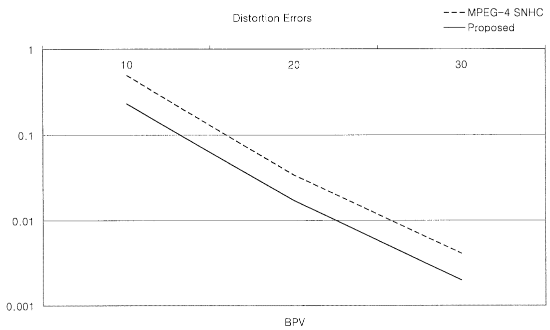
This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultrafast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.



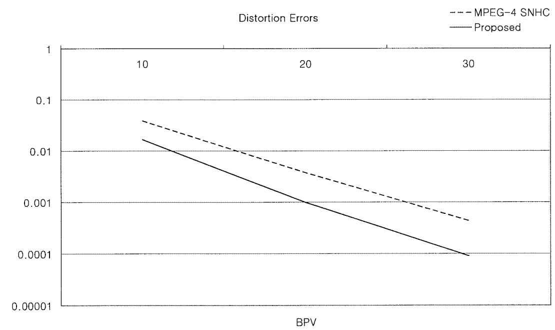
(a) HORSE



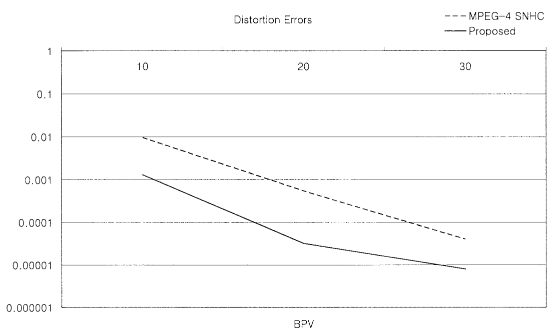
(b) SKULL



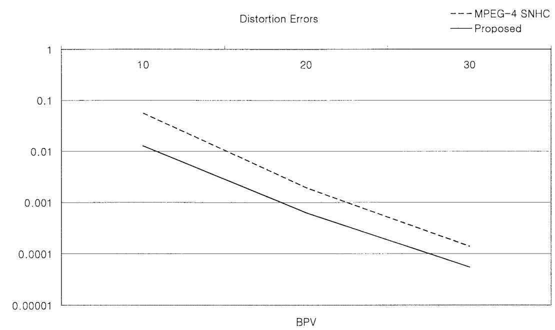
(c) BEETHOVEN



(d) CROCODILE



(e) CAM-SHAFT



(f) 57CHEVY

Fig. 8 Simulation results.

References

[1] J. Hartman and J. Wernecke, The VRML 2.0 Handbook, Addison-Wesley, 1996.

[2] M. Deering, "Geometry compression," Computer Graphics Proceedings, SIGGRAPH 95, pp.13–20, Aug. 1995.

[3] G. Taubin and J. Rossignac, "Geometry compression through topological surgery," ACM Trans. Graphics, pp.84–115, April 1998.

[4] C. Touma and C. Gotsman, "Triangle mesh compression," Proc. Graphics Interfac'98, pp.26–34, 1998.

[5] F. Bossen, On the art of compressing three-dimensional polygonal meshes and their associated properties, Ph.D. Dissertation at EPFL, 1999.

[6] "Description of core experiments on 3-D model coding," ISO/IEC JTC1/SC29/WG11 MPEG98/N244rev1, Atlantic City, Oct. 1998.

[7] J. Li, Progressive compression of 3-D graphics, Ph.D Dissertation at Univ. of Southern California, USA, Aug. 1998.

[8] J.S. Choi, Y.H. Kim, H.J. Lee, I.S. Park, M.H. Lee, and C.T. Ahn, "Geometry compression of 3-D mesh models using predictive two-stage quantization," IEEE Trans. Circuits & Syst. Video Tech., vol.10, no.2, pp.312–322, Dec. 2000.

[9] J.H. Ahn and Y.S. Ho, "Geometry compression of 3-D models using adaptive quantization for prediction errors," Picture Coding Symposium, pp.193–197, April 1999.