Geometry Compression of 3D Models using Adaptive Quantization for Prediction Errors

Jeong-Hwan Ahn and Yo-Sung Ho
Kwangju Institute of Science and Technology
1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea
jhahn@gogh.kjist.ac.kr

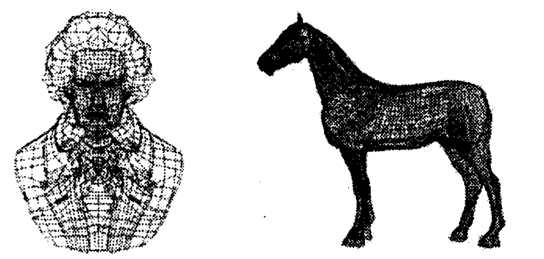
Abstract

The increase of 3D model data requires efficient compression algorithm in order to reduce the cost of data transmission and storage. In general, 3D models are represented by 3D meshes, which are defined by connectivity data, geometry data, and photometry data. In this paper, we address the geometry coding of 3D models. Using the parallelogram prediction, the encoder first predicts vertex positions along a triangle spanning tree. We encode those prediction errors using the uniform quantizer. In some cases, the variance of the prediction errors is large due to a few large error values, which make a visually unacceptable distortion. Therefore, we resolve this problem by a recursive pull-down operation. Since a 3D model is generally composed of multiple components, we also encode root vertices efficiently by exploiting the spatial correlation of neighboring connected components. The proposed scheme demonstrates improved coding efficiency for the selected VRML test data.

1. Introduction

In recent days, the digital interactive multimedia applications using 3D objects are increasing for the internet services and computer graphics, and the MPEG-4 Synthetic/Natural Hybrid Coding (SNHC) [1] group addresses coding of 3D models.

Generally, 3D polygonal models are structured by triangular meshes, which are defined by connectivity data, geometry data, and photometry data, as shown in Figure 1. 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 3D model. The photometry data specify information of each surface such as colors, surface normals, and texture coordinates. Since the geometry data are specified in the 3D space by 3D vectors of three floating-point numbers, it contains a large amount of data. Therefore, it is necessary to represent the geometry information efficiently in order to store, transmit, render, or manipulate 3D data.



(a) BEETHOVEN (b) HORSE

Figure 1. 3D Models

In the geometry compression scheme for 3D triangular meshes proposed by IBM [2,3], 3D coordinate values of each vertex point are initially approximated using an evenly subdivided bounding box along its x, y and z axes, independently. Then, each coordinate value is predicted using a linear combination of previous values along the triangle spanning tree, and the difference of the evenly approximated value and its predicted value is entropy coded. This scheme is simple to implement; however, it generates large quantization errors because the size of the bounding box is large. The quantization error is approximately $\Delta^2/12$, where Δ is the step size of the evenly subdivided bounding box.

In this paper, we describe a new compression scheme for 3D geometry information. Our proposed scheme provides smaller quantization errors than the IBM scheme, since the prediction errors are quantized.

2. A New Scheme for Geometry Compression

A block diagram of the proposed compression scheme is shown in Figure 2. The encoder consists of three stages: preprocessing, uniform quantization, and entropy coder. In the preprocessing stage, the histogram of the prediction errors, Δx, Δy, and Δz, is calculated, respectively. From each histogram, we determine the quantization step size, and apply the uniform quantization over the prediction errors. The quantizer index is then encoded by a QM coder[4].

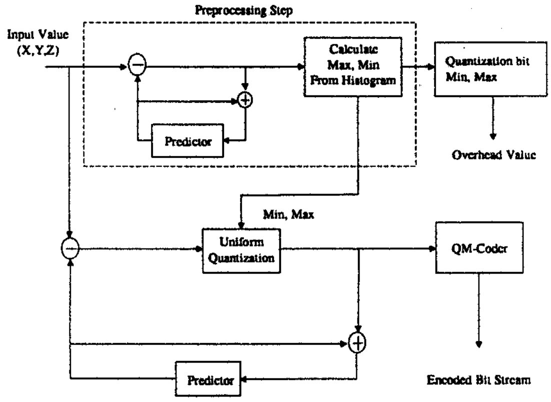


Figure 2. Block Diagram of Geometry Encoder

2.1 Preprocessing

First of all, we examine prediction errors of 3D vertex coordinates from the parallelogram predictor [5] in order to select the optimal bounding values. These values are delivered to the uniform quantizer for the pull-down operation.

2.2 Parallelogram Prediction

We obtain a prediction value $\bar{v}_n = (\bar{x}_n, \bar{y}_n, \bar{z}_n)$ of each vertex point $v_n = (x_n, y_n, z_n)$ based on the parallelogram prediction in the triangle spanning tree [2,3], as illustrated in Figure 3. The parallelogram prediction is adapted for the triangle tree traversal.

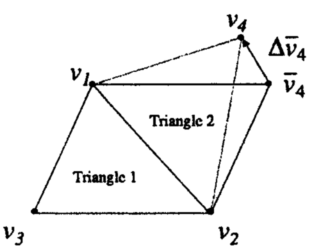


Figure 3. Parallelogram Predictive Rule

When we pass from triangle 1 to triangle 2, the vertices v_1, v_2, and v_3 are already decoded. The opposite vertex v_4 from the common edge (v_1, v_2) is predicted as $v_1 + v_2 - v_3$. Thus, the predicted $\bar{v}_4$, together with its three ancestors, forms a parallelogram and belongs to the same plane. For some vertices, all three traversed ancestors may not be available. If there are only two ancestors, the prediction coefficients are set to 2 and –1. If only one ancestor exists, it is directly used as a prediction value for the current vertex. In case of no ancestors, a null prediction is used.

We then calculate the prediction error Δv_4 by $\Delta v_4 = v_4 - \bar{v}_4$.

2.3 Uniform Quantization and Pull-Down Scheme

We encode the prediction errors using a uniform quantizer. The mean squared quantization error (MSQE) is $\delta^2/12$, where δ is the step size of the evenly subdivided prediction errors. MSQE of the IBM scheme is $\Delta^2/12$, where Δ is the step size for original values. Since δ is smaller than Δ, the proposed scheme produces less MSQE than the IBM scheme.

Occasionally, the variance of the distribution of the prediction errors, Δx, Δy, and Δz, is large due to inadequate prediction. If a 3D model contains many edges and boundaries, it produces some large prediction residues. Although the number of large prediction errors is small, they may generate a visually unacceptable distortion, as shown in Figure 4.

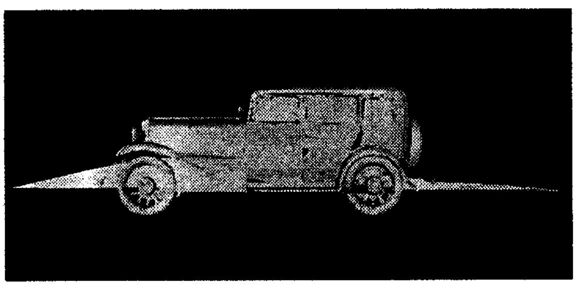


Figure 4. Visual Distribution due to Large Errors

However, we can remedy the problem by a pull-down operation, as explained in Figure 5.

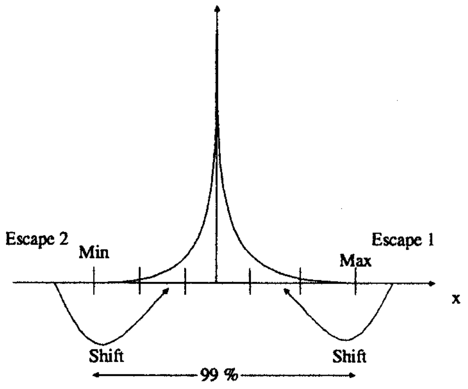


Figure 5. Pull-down Operation

Once we make histogram of the prediction errors, we select maximum bounds within which we have 99% of vertices. If the error value is outside of the maximum bound, the error is subtracted by a maximum value to pull down its magnitude until this value is inside the bound. This process is indicated by the escape code 1 or the escape code 2 in order to recover the correct value at the decoder. It removes the overload errors efficiently.

2.4 Coding of Multiple Components

Usually, a 3D model is composed of multiple components. The first vertex of each component is called as a root vertex. The root vertex plays a very important role as an anchor point for the entire mesh of the connected component. Any change of the root vertex position may cause the crack problem, and the error can be propagated successively. Since the root vertex of each component has no preceding vertices, this root vertex can't be coded in the same way as other vertices. If the root vertex is coded with its own floating point numbers, it is not efficient. Therefore, we propose an adaptive predictor for root vertices [9].

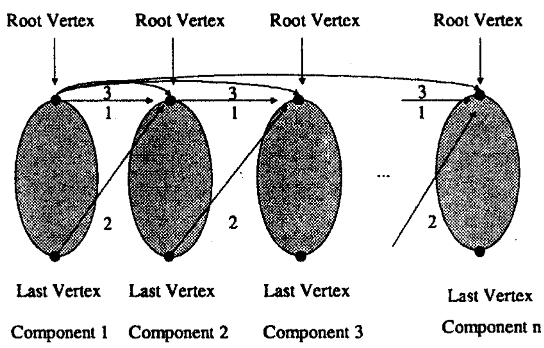


Figure 6. Adaptive Prediction for Root Vertex

As shown in Figure 6, a root vertex can be predicted by the root vertex of the first component, the last vertex of the previous component, or the root vertex of the previous component. Then, the smallest residual value is selected. If the residual value is greater than a predetermined threshold value, the root vertex is directly coded with *n* bits.

Table 1: Flag Bits for Root Coding

Flag	Prediction
00	No prediction
01	Root vertex of first component
10	Last vertex of previous component
11	Root vertex of previous component

As shown in Table 1, two bits are used to represent how the root vertex is predicted.

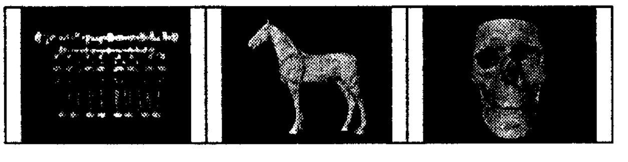
2.5 Entropy Coding

After a modified uniform quantization process, the index values are encoded by a binary arithmetic coder, such as a QM coder[4].

Once we represent the quantization indices as binary symbols, a bit-plane representation [6,7] is possible. Every quantized prediction error is represented by a binary string and encoded by a context-based QM coder based on the 113 state Markov model for the estimated probability [4]. The binary representation of the residues is scanned from the most significant to the least significant bit plane. The context is chosen as the number of neighbors, which have already significant residues at the current quantization stage. If the number of the neighbors is small, the residue is likely to remain insignificant. If the number is large, the residue tends to become significant.

3. Simulation Results

We have tested the performance of our coding algorithm with some Virtual Reality Modeling Language (VRML) [8] models: CAM-SHAFT, HORSE, SKULL, BEETHOVEN, CROCODILE and 57CHEVY, as displayed in Figure 7. The characteristics of the test models are summarized in Table 2.



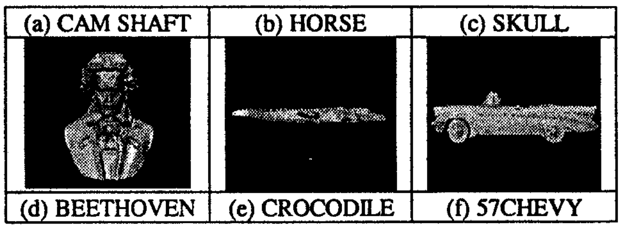


Figure 7. Test Models

We have obtained the prediction errors of vertex positions based on the father-son relationship in the triangle spanning tree generated by EPFL [5].

To evaluate a distortion between the original 3D polygonal model and the reconstructed one, we define the following error metric. The original and the reconstructed models have the same connectivity information, but may have different values of vertex positions. For each vertex of the original model, the closest vertex of a reconstructed model is selected and their distance d_1 is calculated. For each vertex of the reconstructed model, the closest vertex in an original model is determined and their distance d_2 is calculated. The distortion metric is the average value of two asymmetric terms, d_1 and d_2. Therefore, the error measure becomes symmetric with respect to the original and the reconstructed models.

Table 2. Characteristics of Test Models

Model	# of Vertices	# of Components
CAM-SHAFT	54898	209
HORSE	22258	3
SKULL	22104	1
BEETHOVEN	2845	20
CROCODILE	17332	65
57CHEVY	18472	585

Performances of the proposed and the IBM schemes are compared in Table 3 at 10, 20 and 30 bits per vertex (bpv), which shows that the proposed scheme outperforms the IBM scheme.

Table 3. Mean Distortion of Test Models

Model	BPV	Proposed Scheme	IBM Scheme
CAM-SHAFT	10	0.001302	0.008320
	20	0.000032	0.000513
	30	0.000008	0.000032
HORSE	10	0.045341	0.067075
	20	0.003865	0.004221
	30	0.000300	0.000527
SKULL	10	0.051718	0.228374
	20	0.003980	0.014252
	30	0.000355	0.001774
BEETHOVEN	10	0.230881	0.264210
	20	0.017111	0.016763
	30	0.001988	0.002104
CROCODILE	10	0.016640	0.021934
	20	0.000980	0.002757
	30	0.000090	0.000343
57CHEVY	10	0.012737	0.042835
	20	0.000628	0.001313
	30	0.000055	0.000083

4. Conclusion

In this paper, we propose a new coding scheme for 3D geometry information using properties of prediction errors. The proposed method includes the pull-down operation for the large prediction errors to reduce a visually unacceptable noise. For a 3D model with multiple connected components, we applied an adaptive root vertex coding scheme. Experimental results show that our proposed scheme outperforms the bounding box approach.

5. Acknowledgments

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST).

6. References

[1] "SNHC Verification Model 9.0 [3D Mesh Encoding] Draft," ISO/IEC JTC1/SC29/WG11 MPEG98/N2302, Oct. 1998.

[2] G. Taubin and J. Rossignac, "Geometric Compression Through Topological Surgery," ISO/IEC JTC1/SC29/WG11 MPEG98/M3059, Feb. 1998.

[3] G. Taubin, W. Horn and F. Lazarus, "The VRML Compressed Binary Format," ISO/IEC JTC1/ SC29/ WG11 MPEG98/ M3062, Feb. 1998.

[4] W. Pennebacker and J. Mitchell, *JPEG Still Image Compression Standard*, New York: Van National Reinhold, 1993.

[5] "Description of Core Experiments on 3D Model Coding," ISO/IEC JTC1/SC29/WG11 MPEG98/ N2302, July 1998.
[6] J. Li and C. C. Kuo, "Embedded Coding of Mesh Geometry," ISO/IEC JTC1/SC29/WG11 MPEG98/ M3325, March 1998.
[7] J. M. Shapiro, "Embedded image Codin Gusing Zerotrees of Wavelet Coefficients," IEEE Trans. Image Processing, vol. 41, no. 12, pp. 3445-3462, 1993.
[8] J. Hartman and J. Wernecke, *The VRML 2.0 Handbook*, Addosion-Wesley, 1996.
[9] J.H. Ahn and Y.S. Ho, "Adaptive Coding of Multiple Components on 3D Model Coding," ISO/IEC JTC1/SC29/WG11 MPEG98/M4343, Dec. 1998.