COLOR DATA CODING FOR THREE-DIMENSIONAL MESH MODELS CONSIDERING CONNECTIVITY AND GEOMETRY INFORMATION

Young-Suk Yoon [1], Sung-Yeol Kim [2], and Yo-Sung Ho [2]

[1]Electronics and Telecommunications Research Institute (ETRI),
161, Gajeong-dong, Yuseong-gu, Deajeon, 305-350, Korea
[2]Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
ys.yoon@etri.re.kr, {sykim75, hoyo}@gist.ac.kr

ABSTRACT

In this paper, we propose a new predictive coding scheme for color data of three-dimensional (3-D) mesh models. We exploit connectivity and geometry information to improve coding efficiency. After ordering all vertices in a 3-D mesh model with a connectivity coding technique, we propose a geometry predictor to compress the color data efficiently. The predicted color can be obtained by a weighted sum of reconstructed colors for adjacent vertices using both angles and distances between the current vertex and adjacent vertices. Simulation results show that the proposed scheme provides enhanced coding efficiency over previous works for various 3-D mesh models.

1. INTRODUCTION

As broadband convergence networks are widely used, various multimedia services with three-dimensional (3-D) audio-visual data have been developed, such as an augmented reality, immersive media, and realistic broadcasting. Thus, we are dealing with a number of 3-D objects day by day.

3-D objects used in those applications require a tremendous number of bits to represent themselves. In order to store and transmit the 3-D objects, it is reasonable for us to compress 3-D objects to reduce the number of bits for limited storages and network resources.

3-D mesh representation is one of the standard methods to describe 3-D objects, where surfaces of 3-D objects are covered with so many polygons [1]. A 3-D mesh model consists of connectivity, geometry, and photometry data basically. Connectivity data describe incident relations among vertices, but geometry data specify vertex positions in the 3-D space. Photometry data represent colors, normal vectors, and texture coordinates to render 3-D mesh models.
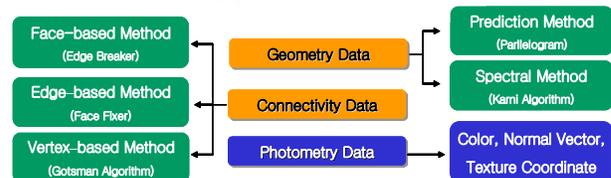
Fig. 1. 3-D mesh coding schemes

Previous methods for the 3-D mesh coding have focused on compressing connectivity and geometry data, as shown in Fig. 1. However, we will focus on the coding scheme for their color data in this paper.

Bajaj et al. [2] adopted a 3-D mesh compression technique for the photometry information using vertex and triangle layers. They used a second-order code predictor to compress geometry and color data.

Ahn and Ho [3] developed a new color coding algorithm based on a mapping table. When the target compression ratio was considerably low, they used a mapping table to encode colors; otherwise, they employed MPEG-4 3-D mesh coding (3DMC) [1] using the differential pulse code modulation (DPCM) scheme [4][5].

Although the previous works claimed good results, there are some rooms for improvement in color data coding. The previous works did not utilize encoded other inforamtion sufficiently. Moreover, they did not consider the properties of 3-D mesh models to eliminate spatial redundancy.

In this paper, we try to exploit color data of adjacent vertices and reflect the characteristics of 3-D mesh models to compress them. We propose a new predictor for colors using connectivity and geometry information of 3-D mesh models.
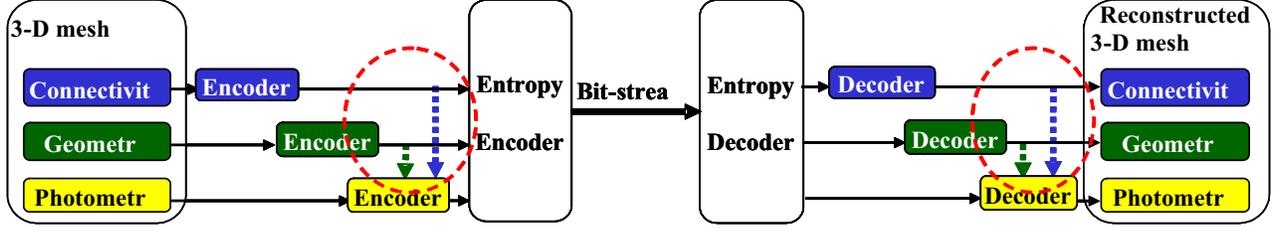
Fig. 2. Configuration of proposed 3-D mesh codec

2. CHARACTERISTICS AND CODEC SYSTEM OF 3-D MESH MODELS

2.1. Mapping Methods

In general, a photometry mapping method can be classified by per-vertex, per-face, and per-corner techniques. In the case of per-vertex binding, photometry data are attached on the vertices of 3-D mesh models; they are shared by all corners around a vertex. For the per-face binding, photometry data are assigned to the faces of 3-D mesh models; they are often shared by all corners around a face. On the occasion of the per-vertex binding, photometry data are appended to the corners of the 3-D mesh models; even though each corner has different photometry information, a common photometry data is usually shared by a set of adjacent corners. In this paper, we propose a color coding scheme based on the per-vertex binding.

2.2. Vertex Type

Before going through the proposed coding schemes for color data, we need to define a current vertex (CV), a previous vertex (PV), a nearest vertex (NV), and an adjacent vertex (AV). CV is a vertex to be coded directly along the order of vertex traversal, while PV is a vertex which have already coded. NV is closest to CV and AVs are vertices connected to CV.

2.3. Proposed 3-D Mesh Codec

Photometry data are encoded with the following procedure. First, vertices are ordered by a connectivity coding algorithm, so that adjacent vertices become consecutive components in the connectivity coding order list. The coding order is usually determined by the connectivity encoder. And then, the position or attribute of each vertex is predicted from those of previous vertices in the ordered list. Finally, the residual errors are quantized and entropy-coded.

We can notice that connectivity, geometry and photometry data are coded independently in a general system including the encoder and decoder (codec) for 3-D mesh models. On the other hand, Fig. 2 shows a proposed 3-D mesh codec in this paper. As shown in Fig. 2, the new 3-D mesh codec uses connectivity and geometry information so as to predict the color data. The dotted line represents the difference between the proposed scheme and previous ones.

3. COLOR DATA CODING

Color is a useful descriptor that often simplifies object identification and extraction from a scene, and humans can perceive thousands of color shades and intensities.

3.1. New Color Coder
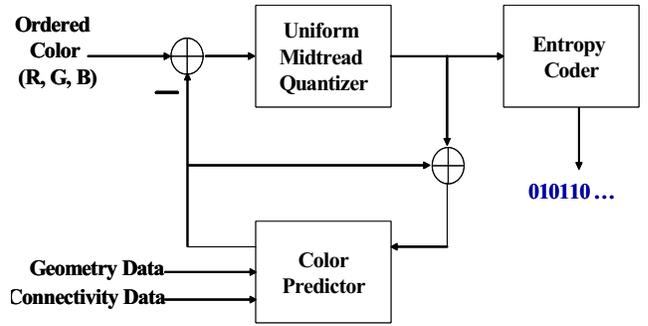
Fig. 3. Proposed color coder

We propose a new scheme to encode the color data in 3-D mesh models. Figure 3 shows the block diagram of the proposed color coder. First, we use a color predictor characterized by a geometry prediction to obtain the predicted color. Then, the residual errors between the original and the predicted colors pass through a uniform midtread quantizer. Finally, the quantized residual error will be coded by an entropy coder [4][5].

At the proposed color coder, we predict the color of CV by a weighted sum of colors of AVs from Eq. (1).

$$color_p(CV) = \sum_{i=1}^{n(AVs)} w_i \times color(AVs_i) \tag{1}$$

In Eq. (1), $color_p(CV)$ is the predicted color of CV, $n(AVs)$ denotes the number of AVs, the term of w_i specifies the weighting factor of the i-th vertex of AVs, and $color(AVs_i)$ represents the reconstructed color of the i-th vertex of AVs.

The sum of the weighting factors should be equal to 1 like Eq. (2).

$$\sum_{i=1}^{n(AVs)} w_i = 1 \qquad (2)$$

3.2. Geometry Prediction

Most compression schemes use the spatial redundancy on the image or video [6]. Like the proceeding, we assume that the color data for 3-D mesh models also contain the spatial redundancy in this paper.

We should largely consider two points to apply traditional coding techniques to 3-D mesh models in order to develop an efficient coding scheme. First, the vertices of 3-D mesh models are arbitrarily located on the 3-D space unlike 2-D images. Next, a pixel always has eight neighboring pixels at fixed positions but for pixels on the border of a 2-D image. However, a vertex does not have the constant number of adjacent vertices and it has various incident relations between itself and adjacent vertices except a regular mesh model. It is accordingly hard to find a clue so as to encode color data considering adjacent vertices.

There are many algorithms using distance, since a predicting pixel value is highly related with both near and adjacent pixel values in the 2-D image. To give the latest instance, a linear 'plane' function exploit the concept of distance at intra 16x16 mode of H.264 which is one of the video coding standard.

In order to reduce spatial redundancy, we assume that color data are in inverse proportion to Euclidean distance, obtaining the reconstructed color information on the color coder [7].

The first relation is expressed by

$$color_p(CV) \propto \frac{1}{d(CV, AVs_i)} \qquad (3)$$

where $d(CV, AVs_i)$ denotes the Euclidean distances between CV and AVs.
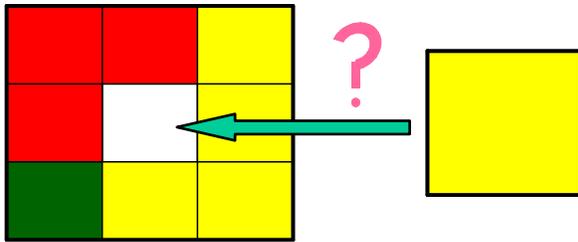
Fig. 4. Influence of neighboring pixels

We have known traditional context-based coding methods to obtain the best coding efficiency [4][5]. In case of a 2-D image, the color data of a pixel can be affected by the neighboring pixels. As we can see in Fig. 4, we can predict the color of neighboring pixels with the best influence. Considering context-based coding algorithms as has been mentioned, we suggest that the predicted color be obtained by a weighted sum of colors of adjacent vertices using angles between CV and AVs [8].
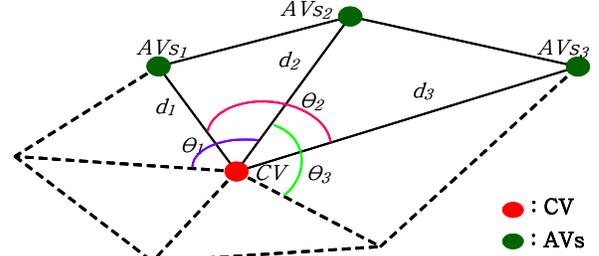
Fig. 5. Example of geometry prediction

The second relation is represented by

$$color_p(CV) \propto \theta_i \qquad (4)$$

where θ_i represents the sum of angles affected by a predicted color data of CV. Therefore, we can define θ_i in the form

$$\theta_i = \angle AVs_{(i-1)\%n(AVs)} CV\ AVs_i + \angle AVs_i CV\ AVs_{(i+1)\%n(AVs)} \qquad (5)$$

where AVs_i denotes one of AVs and % defines a modulo function in Fig. 5.

In general, we can get both distances and angles by using connectivity and geometry information. Thus, we develop that the predicted color for CV can be obtained by a weighted sum of colors of AVs considering both distances and angles between CV and AVs called by a geometry prediction. In hence, we obtain the following relation between CV and AVs with Eq. (3) and Eq. (4).

$$color_p(CV) \propto \frac{\theta_i}{d(CV, AVs_i)} \qquad (6)$$

From Eq. (2) and Eq. (6), the weighting factor w_i can be obtained by the geometry prediction as follows.

$$w_i = \frac{\dfrac{\theta_i}{d(CV, AVs_i)}}{\displaystyle\sum_{j=1}^{n(AVs)} \dfrac{\theta_j}{d(CV, AVs_j)}} \qquad (7)$$

As a result, we can design the color predictor to offer more efficient coding efficiency with Eq. (1) and Eq. (7). The predicted color for CV is expressed by

$$color_p(CV) = \frac{\displaystyle\sum_{i=1}^{n(AVs)} \dfrac{\theta_i}{d(CV, AVs_i)} \times color(AVs_i)}{\displaystyle\sum_{j=1}^{n(AVs)} \dfrac{\theta_j}{d(CV, AVs_j)}}. \qquad (8)$$

4. EXPERIMENTAL RESULTS

In order to evaluate the coding efficiency of the proposed scheme, we compare the results of the proposed algorithm with those of the existing scheme used in MPEG-4 3DMC.

(a) GLOBE **(b) NEFERTITI**

(c) SPHERE **(d) TAL**
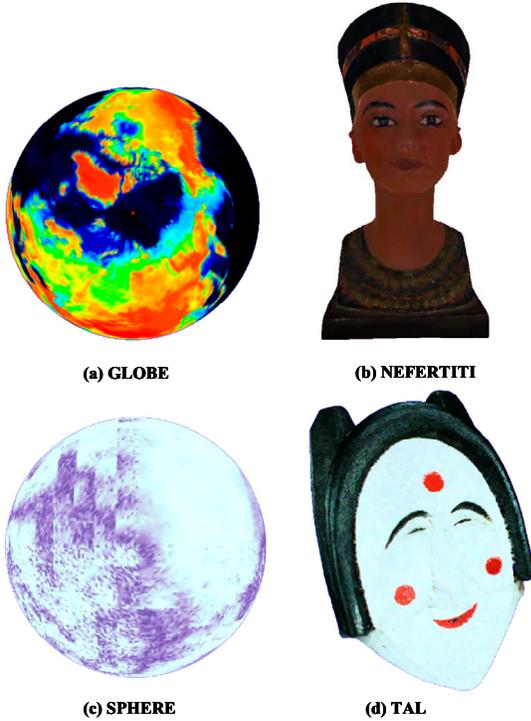
Fig. 6. Test Models for color coding

Figure 6 illustrates the test models for color coding used to compare MEPG-4 3DMC based on DPCM scheme with the proposed algorithm using geometry prediction.

Table 1 lists properties of test models in the color coding. The term of $n(F)$ is defined by the number of the set of faces F, $n(C)$ denotes the number of the set of colors C, and $n'(C)$ indicates the number of the colors without duplication. Since we employ a coding scheme for color data as per-vertex binding, $n(V)$ is equal to $n(C)$ and is lager than $n'(C)$.

In Table 2, we compare coding performances of the proposed algorithm with those of the MPEG-4 3DMC in the condition of the similar color distortion. The coding performances represent compressed sizes for color data of 3-D test models when the quantization level is 24 bits per color (bpc). Since the proposed geometry prediction uses connectivity and geometry information, it is a bit more complicated than previous works. However, it outperforms the MPEG-4 3DMC algorithm, as shown in Table 2. Therefore, we note that the proposed geometry prediction is efficient for color coding.

Table 1. Properties of test models

	GLOBE	NEFERTITI	SPHERE	TAL
$n(V)$	36,866	10,013	41,369	30,737
$n(F)$	73,728	20,022	82,734	61,470
$n(C)$	36,866	10,013	41,369	30,737
$n'(C)$	5,562	7,943	337	8,412

Table 2. Coding efficiency comparison of color coding

	GLOBE	NEFERTITI	SPHERE	TAL
MPEG-4 3DMC	9.80	15.63	18.98	14.09
Proposed scheme	9.55	14.36	17.85	13.22

5. CONCLUSIONS

In this paper, we proposed an efficient coder for color data based on geometry prediction. In order to reduce the spatial redundancy in 3-D mesh models, we took it for granted that adjacent vertices have very similar color information. Moreover, we took into account 3-D mesh model information dependently. Thus, we developed geometry prediction to compress the color data and exploited the predictive coding scheme for color coding with improved coding efficiency. Experimental results have demonstrated that the proposed coding technique is superior to the MPEG-4 3DMC standard for various test models.

6. REFERENCES

[1] "Description of core experiments on 3D model coding," ISO/IEC JTC1/SC29/WG11 MPEG/M3325, 1998.

[2] V. Bajaj, V. Pascucci and G. Zhuang. "Single Resolution Compression of Arbitrary Triangular Meshes with Properties," *Proceedings of the Data Compression Conference*, pp. 167-186, Mar. 1999.

[3] J.H. Ahn, C.S. Kim, and Y.S. Ho, "An Efficient Coding Algorithm for Color and Normal Data of Three-dimensional Mesh Models," *Proceedings of International Conference on Image Processing*, pp. 789-792, Sept. 2003.

[4] K. Sayood, *Introduction of Data Compression*, Morgan Kaufmann, 2000.

[5] D. Saloman, *Data Compression : The Complete Reference*, Springer Verlag, 2004.

[6] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989

[7] Y.S. Yoon, S.Y. Kim, and Y.S. Ho, "Predictive Coding of Photometry Information of Three-dimensional Meshes Using Geometry and Connectivity Information," *Proceedings of International Workshop on Advanced Image Technology*, pp. 7-12, Jan. 2005.

[8] Y.S. Yoon, S.Y. Kim, and Y.S. Ho, "Photometry Data Coding for Three-dimensional Mesh Models Using Connectivity and Geometry Information," *LNCS 3768*, pp. 794-805, Nov. 2005.