Photometry Data Coding for Three-Dimensional Mesh Models Using 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
ys.yoon@etri.re.kr

[2] Gwangju Institute of Science and Technology (GIST),
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
{sykim75, hoyo}@gist.ac.kr

Abstract. In this paper, we propose new predictive coding schemes for photometry data of three-dimensional (3-D) mesh models as per-vertex binding. We exploit geometry and connectivity information to enhance coding efficiency of the color and normal vector data. For color coding, we predict the color of the current vertex by a weighted sum of colors of adjacent vertices considering angles between the current vertex and the adjacent vertices. For normal vector coding, we generate an optimal plane using distance equalization to predict the normal vector of the current vertex. Experimental results show that the proposed coding schemes provide improved coding performance over previous works for various 3-D mesh models.

Keywords: Color coding, normal vector coding, photometry coding, 3-D mesh compression.

1 Introduction

As applications of three-dimensional (3-D) computer graphics become more popular, we handle a large number of 3-D objects that are created by many digital contents providers. In order to keep and deliver 3-D objects more efficiently in the multimedia processing system being able to include digital contents distribution structure [1], we need to compress 3-D objects to reduce the number of bits for storage and transmission.

The 3-D mesh model is one of the standard methods to represent 3-D objects, where the 3-D surface is covered by polygons. In general, a 3-D mesh model can be constructed by combining three major information: connectivity, geometry, and photometry data [2]. While geometry data specify vertex locations in the 3-D space, connectivity data describe the incidence relationship among vertices. Photometry data, which are colors, surface normal vectors, and texture coordinates, are required to paint and shade 3-D mesh models. In this paper, we focus on photometry data coding of the 3-D mesh model as per-vertex binding.

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part II, LNCS 3768, pp. 794–805, 2005.

Bajaj *et al.* [3] developed a 3-D mesh compression method for the photometry information using vertices and triangle layers. They employed a second-order predictor to compress geometry information and color data. In order to predict the normal vector of current vertex, they calculated the average of normal vectors for incident faces. However, they did not exploit the coded information for the photometry data coding. Especially, their color coding method was identical to the geometry coding algorithm.

Ahn *et al.* [4] employed a mapping table for color coding, when the target compression ratio was not high. Otherwise, they adopted the MPEG-4 3-D mesh coding (3DMC) method [2]. However, their method is not suitable for 3-D mesh models of high quality when the number of colors is large. They also developed a compression scheme for normal vectors using the average prediction and the 6-4 subdivision [4]. Although their work produces good results for even meshes, there are some rooms for improvement for uneven meshes.

In this paper, we try to utilize photometry data of adjacent vertices and reflect characteristics of 3-D mesh models. We have proposed new predictors for colors and normal vectors using geometry and connectivity information of the 3-D mesh model. This paper is organized as follows. At first, we talk about several types of vertex in Section 2. Section 3 explains our proposed color coding scheme, and Section 4 describes the proposed normal vector coding method. After providing experimental results in Section 5, we conclude in Section 6.

2 Vertex Classification

In order to explain the proposed coding schemes for photometry data, we need to define a current vertex (CV), a previous vertex (PV), a nearest vertex (NV), and adjacent vertices (AVs) [5]. CV is a vertex to be coded directly along the order of vertex traversal, but PV is a vertex which have already coded. NV is closest to CV, and AVs are the set of vertices connected to CV.

In general, previous works employed only the PV to predict the photometry data of CV. However, since either NV or AVs have very similar photometry data for CV, it is inefficient to use only the PV as the photometry data coding. When we examined the case that the photometry data of NV was analogous to those of AV, the probability was about 99%. Also, the probability was approximately 18% when a vertex that was both one of AVs and PVs contained the same photometry data as CV. Moreover, the probability was around 82% while CV possessed the similar photometry data to not PV but one of AVs. With considering the simple tests mentioned above, we can conclude that we should consider the photometry data of AVs as well as PV as the photometry data coding.

3 Color Coding

3.1 Color Coder

Since angles between CV and AVs influence on the color data of CV, we can develop an angle prediction method for color coding of the 3-D mesh model.

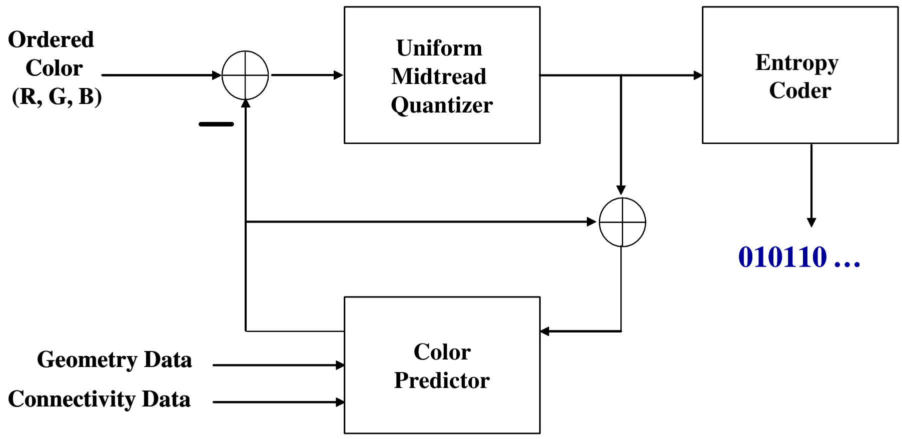

Fig. 1. Block diagram of the proposed color encoder

Figure 1 shows the block diagram of the proposed color coder. First, we use a color predictor characterized by the angle prediction to calculate the predicted color. Then, the difference between the original and the predicted colors goes into the an uniform midtread quantizer. Finally, the quantized residual error is coded by the entropy coder.

In the proposed color coder, we predict the color of CV by a weighted sum of colors of AVs, i.e.,

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

where $color_p(CV)$ is the predicted color of CV, $n(AVs)$ represents the number of AVs, w_i depicts the weighting factor of the i^{th} vertex of AVs, and $color(AVs_i)$ denotes the reconstructed color of the i^{th} vertex of AVs. In Eq. 1, the sum of the weighting factors should be equal to 1.

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

3.2 Angle Prediction

Most image compression algorithms are based on the inspection that for any randomly chosen pixel in the 2-D image, its near neighboring pixels tend to have the very similar value to the pixel. A context-based image compression technique universalize this inspection. It is founded by the intuition that the context of a pixel can be used to estimate the probability of the pixel. Thus the context-based coding technique which is generally used in image compression supports very good coding performance. It predicts a pixel value exploiting coded values of

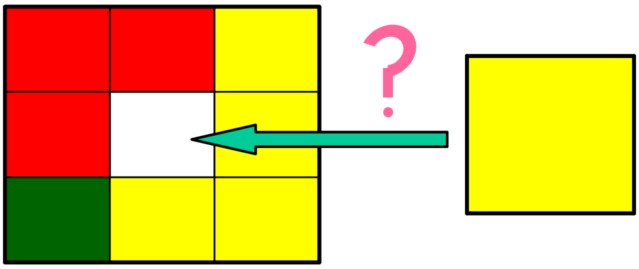

Fig. 2. Simple example of context-based coding method

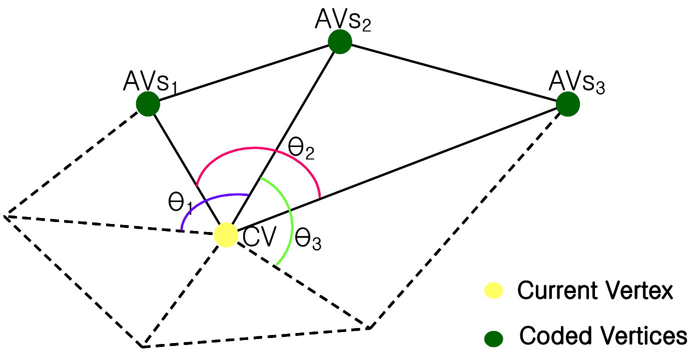

Fig. 3. Example of angle prediction

neighboring pixels [6][7][8]. Figure 2 depicts the uncomplicated case of context-based coding technique which reckons coding pixel value as the most occurring coded value of adjacent pixels.

We should consider two essential respects for employing traditional context-based coding technique in the 2-D space as compressing 3-D mesh models. First, vertices of a 3-D mesh model exist any place on the 3-D space, unlike pixels of 2-D image. Second, pixels always have eight neighboring pixels except for boundary pixels but vertices do not possess the fixed number of AVs. Moreover the mesh topology is so various. Hence, it is very difficult to find the context for color coding from AVs in the 3-D space. In this paper, considering angles between CV and AVs, we can predict the color value of CV by a weighted sum of colors of AVs. This is called an *angle prediction*:

$$color_p(CV) \propto \theta_i \tag{3}$$

where we define θ_i as the sum of angles affected by the color data of AVs_i.

Figure 3 illustrates an example of the angle prediction. CV is the current vertex and AVs_i is one of AVs. As shown in Fig. 3, we can represent θ_i as

$$\theta_i = \angle AVs_{(i-1)\%n(AVs)}CVAVs_i + \angle AVs_iCVAVs_{(i+1)\%n(AVs)} \tag{4}$$

where % denote the modulo operation.

First of all, we can produce the weighting factor w_i with Eq. 2, Eq. 3, and Eq. 4:

$$w_i = \frac{\theta_i}{\sum_{j=1}^{n(AVs)} \theta_j}. \tag{5}$$

Finally, we can predict the color data of CV using the angle prediction from Eq. 1 and Eq. 5:

$$color_p(CV) = \frac{\sum_{i=1}^{n(AVs)} \theta_i \times color(AVs_i)}{\sum_{j=1}^{n(AVs)} \theta_j}. \tag{6}$$

4 Normal Vector Coding

4.1 Normal Vector Encoder

Since previous works [2][3][4] do not consider the normal vector of the opposite side and simply use the normal vectors of non-AVs which are not highly correlated with CV, they fail to obtain a good prediction of the original normal vector. However, our proposed scheme for normal vector coding exploits all the available geometry and connectivity information of the 3-D mesh model. The main idea of our normal vector coding scheme is to generate an optimal plane using distance equalization for normal vector prediction [5].

Figure 4 shows the block diagram of the normal vector coder with an normal vector predictor. Then, we transform the Cartesian coordinate system into the

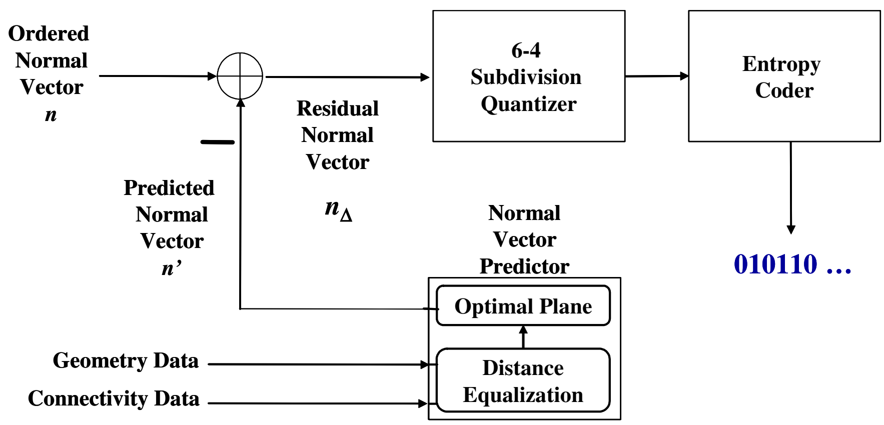

Fig. 4. Block diagram of the proposed normal vector encoder

spherical coordinate system with the unit radius, we can obtain the predicted normal vector and the residual normal vector. Then, we use the 6-4 subdivision quantizer to quantize the residual normal vector and encode the quantizer index by the QM coder [4].

4.2 Optimal Plane Using Distance Equalization

Optimal Plane. In order to predict the normal vector of CV, we generate an optimal plane with the minimum square error from AVs. The optimal plane for CV is obtained from AVs using the least squares approximation (LSA) method [9].

Figure 5 describes the generation of the optimal plane for CV. Figure 5(a) represents the plane view of optimal plane and Figure 5(b) depicts the side view of optimal plane. As shown in Fig. 5, the optimal plane of CV may not include the all of the AVs for CV.

When the optimal plane has the shortest average distance from AVs, the equation of the optimal plane can be expressed by

$$ax + by + cz + d = 0 \tag{7}$$

where the normal vector of the optimal plane, $\boldsymbol{n}_{op} = (a, b, c)$ and $\|\boldsymbol{n}_{op}\| = 1$. If $a \neq 0$, Eq. 7 can be rewritten as

$$\frac{b}{a}y + \frac{c}{a}z + \frac{d}{a} = -x, a \neq 0. \tag{8}$$

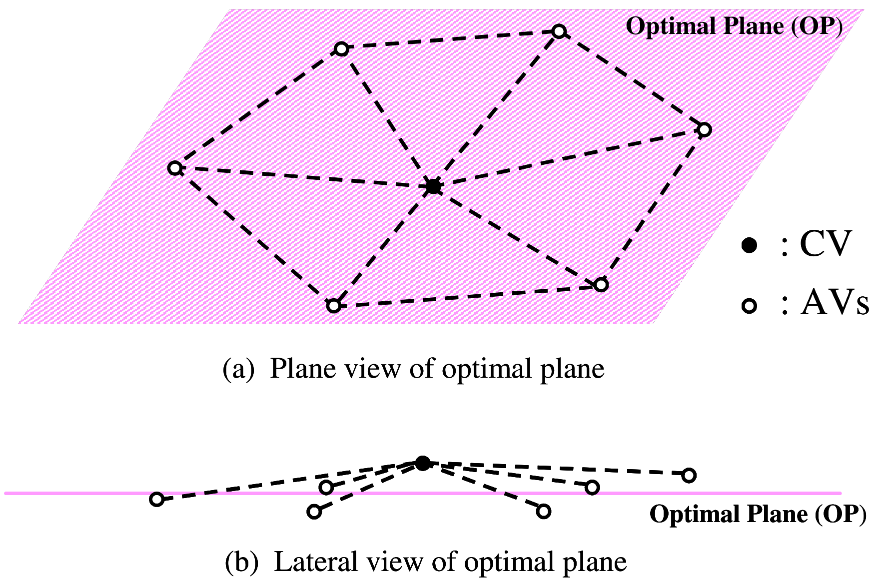

Fig. 5. Optimal plane

Moreover, the coordinate of the i^{th} AV can be represented by (x_i, y_i, z_i). We can write down the equations which would hold if the optimal plane could go through all AVs:

$$\begin{bmatrix} 1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ 1 & y_{n(AVs)} & z_{n(AVs)} \end{bmatrix} \begin{bmatrix} \frac{d}{a} \\ \frac{b}{a} \\ \frac{c}{a} \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n(AVs)} \end{bmatrix} \wedge Ax = b \tag{9}$$

When Eq. 9 has a solution, there would be no minimum square error and AVs would be on the optimal plane. However, since the number of AVs is generally greater than 3, the matrix A is $n(AVs) \times 3$ and we can obtain $\boldsymbol{n}_{op}$ by the least squares approximation.

Let $\overline{x}$ be a solution to Eq. 9; thus $A\overline{x}$ is the closest point to b. In this case, the difference $A\overline{x} - b$ must be a vector orthogonal to the column space of A. This means that $A\overline{x} - b$ is perpendicular to each of the columns of A, and hence $A^T(A\overline{x} - b) = 0$. Multiplying and separating terms gives an equation

$$A^T A\overline{x} = A^T b. \tag{10}$$

Although A is not of full rank, this set of equations should have a solution, since both the columns of A are linearly independent and $A^T b$ lies in the column space of $A^T A$. The matrix $A^T A$ is invertible, and so $\overline{x}$ may be found by

$$\overline{x} = (A^T A)^{-1} A^T b. \tag{11}$$

Thus we can obtain $\boldsymbol{n}_{op}$ with Eq. 11 and $\|\boldsymbol{n}_{op}\| = 1$.

Distance Equalization. If a 3-D mesh model is regular and even, the optimal plane provides a good prediction for the normal vector. However, there are some rooms for improvement in other cases. In order to obtain a better prediction for the normal vector, we propose a distance equalization technique that adjusts the distance between CV and each of the AVs be the same.

Figure 6 describes the difference between the method adopting only the optimal plane and the approach employing optimal plane applied to distance equalization technique. Figure 6(a) shows a simple example of the 3-D mesh model that has similar dihedral angles, i.e., $\theta_i \cong \theta_j$. As a result, the predicted normal vector of CV is equal to the normal vector of the optimal plane. Figure 6(b) illustrates a general case having different dihedral angles, i.e., $\theta_i \neq \theta_j$. In this case, we have an error when substituting the normal vector of CV by the normal vector of the optimal plane. We apply the distance equalization to AVs to reduce the prediction error of the normal vector of CV. With distance equalization, we obtain very similar dihedral angles, i.e., $\theta_i' \cong \theta_j'$. Since distance equalization is intrinsic to the characteristics of an isosceles triangle, we can obtain accurate predicted normal vectors.

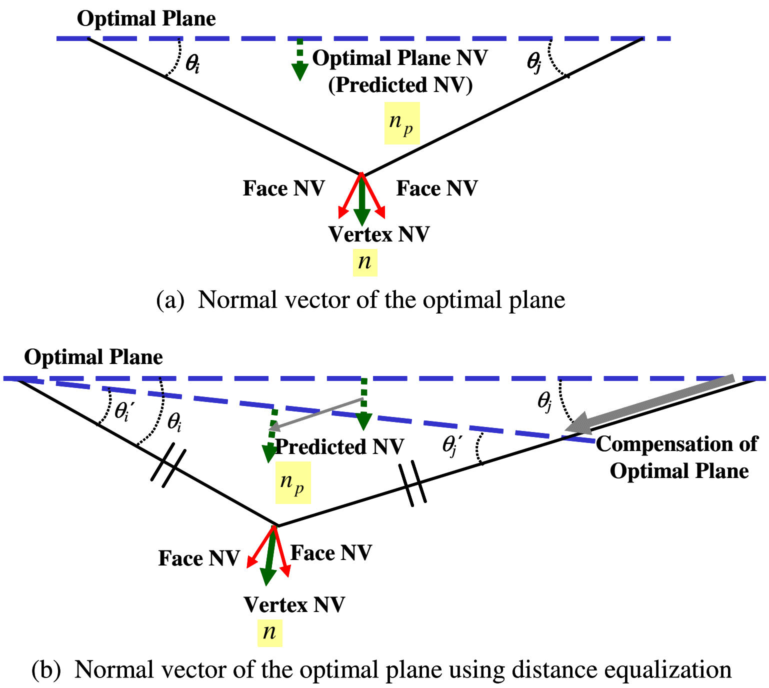

Fig. 6. Distance equalization

5 Experimental Results

In order to evaluate the performance of the proposed methods, we compare experimental results of the our schemes with those of the existing algorithms.

5.1 Results of Color Coding

Figure 7 shows 3-D mesh models used for performance evaluation of color coding. Table 1 lists properties of test models for color coding. We define that $n(F)$ be 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 $n(C)$.

Table 1. Test models for color coding

	GLOBE	NERFERTITI	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

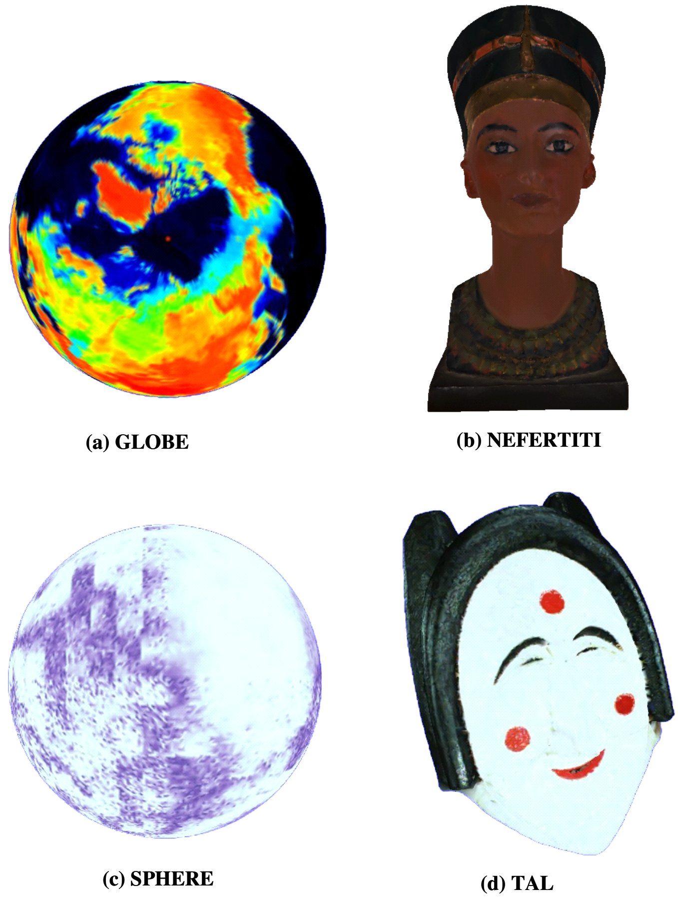

Fig. 7. Test models with color data

Table 2. Performance comparison of color coding

	GLOBE	NEFERTITI	SPHERE	TAL
MPEG-4 3DMC	9.80	15.63	18.98	14.09
Angle Prediction	9.79	14.21	17.83	13.18

In Table 2, we compare coding performances of the proposed algorithm with those of the MPEG-4 3DMC algorithm [2] when they have 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). As shown in Table 2, the proposed scheme outperforms the MPEG-4 3DMC algorithm. Hence, we note that the proposed angle prediction is efficient for color coding.

5.2 Results of Normal Vector Coding

Figure 8 shows 3-D mesh models used for comparison on coding efficiency of normal vector. In this simulation, we included the 'CROCODILE' and 'HORSE' models instead of using the 'GLOBE' and 'NEFERTITI' models so as to check the coding performance with respect to uneven mesh models.

Table 3 shows properties of test models for normal vector coding. We define $n(N)$ as the number of normal vectors N. Owing to per-vertex binding, $n(V)$ is equal to $n(C)$. Furthermore, the amount of uncompressed normal vectors is equal to the amount of uncompressed colors and the amount of uncompressed geometry information.

A proper measure for distance between two normal vectors is the angular distortion that is defined by

$$d_{normal} = \frac{1}{n(V)} \sum_{i=1}^{n(V)} \arccos(n_i \cdot n_i') \qquad (12)$$

where n_i and n_i' represent the source and reconstructed normal vectors of the i^{th} vertex, respectively.

Table 4 compares coding performances of the proposed scheme employing the optimal plane using distance equalization to previous works when the quantization level is 15 bits per normal vector. From Table 4, we can conclude that the proposed scheme produces less angular distortion than the existing algorithms.

(a) CROCODILE (b) HORSE

Fig. 8. Test models with normal vector data

Table 3. Test models for normal vector coding

	CROCODILE	HORSE	SPHERE	TAL
$n(V)$	17,332	19,851	41,369	30,737
$n(F)$	34,404	39,698	82,734	61,470
$n(N)$	17,332	19,851	41,369	30,737

Table 4. Comparison of coding efficiency for normal vectors

	CROCODILE	HORSE	SPHERE	TAL
MPEG-4 3DMC	0.88	0.36	0.05	0.26
Average Prediction	0.82	0.30	0.03	0.24
Proposed scheme	0.38	0.06	0.004	0.06

Thus, the proposed scheme employing the optimal plane using distance equalization contributes to efficient coding of normal vectors for uneven mesh models as well as even mesh models.

6 Conclusions

In this paper, we have proposed new coding schemes of photometry data using geometry and connectivity information of the 3-D mesh model. With considering the spatial correlation among those information, we can develop improved predictive coding schemes for colors and normal vectors. For color coding, we have developed a new prediction algorithm for the color of the current vertex by considering the angles between the current vertex and adjacent vertices. For normal vector coding, we proposed to form an optimal plane using distance equalization. Experimental results have demonstrated that the proposed coding methods outperform previous works for various test models.

Acknowledgements. This work was supported in part by Electronics and Telecommunications Research Institute (ETRI), in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

1. Lee, J.S., Hwang, S.O., Jeong, S.W., Yoon, K.S., Park, C.S., and Ryou, J.C. : A DRM Framework for Distributing Digital Contents through the Internet. ETRI Journal (2003) 423-436
2. ISO/IEC JTC1/SC29/WG11 MPEG/M3325 : Description of core experiments on 3D model coding (1998)
3. Bajaj, V., Pascucci, V., and Zhuang, G. : Single Resolution Compression of Arbitrary Triangular Meshes with Properties. Proceedings of the Data Compression Conference (1999) 167-186
4. Ahn, J.H., Kim, C.S., and Ho, Y.S. : An Efficient Coding Algorithm for Color and Normal Data of Three-dimensional Mesh Models. Proceedings of International Conference on Image Processing (2003) 789-792

5. Yoon, Y.S., Kim, S.Y., and Ho, Y.S. : Predictive Coding of Photometry Information of Three-dimensional Meshes Using Geometry and Connectivity Information. Proceedings of International Workshop on Advanced Image Technology (2005) 7-12
6. Sayood, K. : Introduction of Data Compression, Morgan Kaufmann (2000)
7. Salomon, D. : Data Compression : The Complete Reference. Springer Verlag (2004)
8. Jain, A.K. : Fundamentals of Digital Image Processing. Prentice-Hall (1989)
9. Strang, G. : Linear Algebra and Its Applications. Tomson Learning (1986)