New methods for progressive compression of colored 3D Mesh

Ho Lee
Université de Lyon, CNRS
Université Lyon 1, LIRIS,
UMR5205, F-69622, France
ho.lee@liris.cnrs.fr

Guillaume Lavoué
Université de Lyon, CNRS
INSA-Lyon, LIRIS,
UMR5205, F-69621, France
guillaume.lavoue@liris.cnrs.fr

Florent Dupont
Université de Lyon, CNRS
Université Lyon 1, LIRIS,
UMR5205, F-69622, France
florent.dupont@liris.cnrs.fr

ABSTRACT

In this paper, we present two methods to compress colored 3D triangular meshes in a progressive way. Although many progressive algorithms exist for efficient encoding of connectivity and geometry, none of these techniques consider the color data in spite of its considerable size. Based on the powerful progressive algorithm from Alliez and Desbrun [All01a], we propose two extensions for progressive encoding and reconstruction of vertex colors: a prediction-based method and a mapping table method. In the first one, after transforming the initial RGB space into the Lab space, each vertex color is predicted by a specific scheme using information of its neighboring vertices. The second method considers a mapping table with reduced number of possible colors in order to improve the rate-distortion tradeoff. Results show that the prediction method produces quite good results even in low resolutions, while the mapping table method delivers similar visual results but with a fewer amount of bits transmitted depending on the color complexity of the model.

Keywords: Progressive compression; Colored 3D mesh.

1. INTRODUCTION

Nowadays, 3D models are widely used in many applications such as virtual reality, entertainment, Computer-Aided Design, scientific simulation and e-commerce. Among the various existing representations, 3D triangular meshes are particularly appropriate to represent these models due to their algebraic simplicity so that the most part of manipulations can be processed by the graphic hardware. The increasing popularity and the increasing size of 3D meshes to respond to the needs of representing objects or scenes with more and more realism have become a critical issue, especially for the end-users with limited bandwidth and storage capacity. In this context, compression is a good solution for this task; two different classes of techniques exist: *single-rate* and *progressive*. Single-rate techniques compress the mesh information as a whole and the visualization is possible only when the entire compressed file is received at the user-side. These techniques often have advantages in terms of compression ratio. On the other hand, progressive techniques are more flexible by providing the possibility of early visualization of the coarse version with very few bits transmitted and then more refined models can be rendered when more bits are received. This property of progressive reconstruction is useful especially for large models and for Internet-based applications.

A typical 3D mesh is composed by its geometry, connectivity and attribute data. Geometry data determine vertex positions in the 3D space. Connectivity data describe how these vertices are connected together and attribute data specify colors, surface normals or texture information for instance.

Among these mesh elements, attribute data is not often considered by the state-of-the-art mesh compression algorithms in spite of their visual importance and their considerable size, especially for the progressive algorithms.

In this paper, we propose two approaches to encode efficiently color data in a progressive manner. Our work can be seen as an extension of the progressive mesh compression algorithm from [All01a] which encodes only the connectivity and the geometry. We have chosen this algorithm, since it is the best state-of-the-art connectivity-driven algorithm. As it was observed in [Lee09], even the most efficient geometry-guided algorithm [Pen05] produces a poor visual quality at low and medium bit rate, due to the stair-like effects. Moreover, Alliez and Desbrun's algorithm which is based on the vertex removal allows a better prediction using more neighboring vertices than algorithms based on edge-contraction [Hop96] [Paj00] [Tau98a] [Kar02], leading to the better compression of the color data.

Related work

Single-rate techniques have been firstly studied by many researchers in order to reduce compactly the mesh data [Tau98b] [Tou98] [Gum98] [Baj99] [Ros99] [All01b].

Later on, research on progressive compression techniques have been introduced with the increasing popularity of web-based applications. The first progressive algorithm was proposed by Hoppe [Hop96]. This new mesh representation, *progressive mesh*, simplifies a given mesh by applying successively edge contraction operations. At each step, the edge to be contracted is properly chosen in order to reduce the approximation error as much as possible. At the decompression stage, the reconstruction is achieved by the inverse operation, vertex split. This method has been extended by several researchers to improve the compression efficiency and also the rate-distortion trade-off [Paj00] [Tau98a] [Kar02]. In their work, Cohen-Or et al. [Coh99] proposed the patch coloring algorithm for progressive transmission. This algorithm removes iteratively an independent vertex set – any two vertices of this set are not connected by an edge – using vertex decimation. Then, each hole left by vertex decimation is re-triangulated in a deterministic way. The set of these new triangles is called a patch. The authors applied 2-coloring and 4-coloring methods to the patches in order to permit the decoder to identify correctly each patch. This algorithm encodes the connectivity with an average of 6 bits-per-vertex (bpv). Alliez and Desbrun [All01a] extended the existing valence-driven single-rate approaches [Tou98] [All01b] for progressive mesh encoding. Their algorithm, which is also based on vertex decimation, consists of two conquests: decimation and cleansing. The decimation conquest is successively applied alternating with cleansing conquest, building different levels of details. This algorithm encodes the connectivity with an average of 3.7 bpv.

All the progressive algorithms described above are connectivity-driven algorithms, meaning that the priority is given to the connectivity coding. Observing that the amount of geometry data in the compressed file is often larger than connectivity data, Gandoin and Devillers [Gan02] proposed the first geometry-driven approach based on the kd-tree space subdivision. In terms of lossless compression ratio, this algorithm outperforms connectivity-driven algorithms. Peng and Kuo [Pen05] proposed a more efficient geometry-guided technique by using the octree cell subdivision. An improvement is achieved by using efficient prediction methods for both connectivity and geometry. These geometry-driven algorithms give very impressive results in terms of lossless compression ratio, however they provide quite poor results at low resolutions, hence they are not fully efficient for progressive transmission. In [Lee08], the authors proposed key-frame based technique for the efficient transmission of animating meshes.

Up to present, the compression of the mesh attribute data such as colors, normals or texture coordinates plays a secondary role. Among the well-known single-rate techniques, only [Dee95] [Baj99] [Tau98b] proposed a method to encode vertex-bind color information in the RGB color space. However, the prediction and the quantization used for the color encoding are the same as for the geometry encoding regardless of its different nature. More recently, Ahn et al. [Ahn06] and Yoon et al. [Yoo07] proposed new methods for the efficient encoding of color data. Ahn et al. [Ahn06] used a mapping table based on the vertex layer traversal algorithm. Instead of encoding color coordinates of each vertex, they encode the index of the vertex color in the mapping table. A color value in the mapping table is encoded when it appears for the first time during the traversal. In other words, they have to encode the index of each vertex and the corresponding color coordinates in the mapping table. To further improve the efficiency, they also used a delta coding for color index encoding. Yoon et al. [Yoo07] introduces a prediction method using connectivity and geometry information of neighboring vertices. They consider different weights for the neighboring vertices using angle analysis. Then the color value of the current

vertex is predicted from weighted averaged color values.

Geometry images [Gu02] [Yao08] permit to represent compactly the colored geometric models using 2D images. There exist also some algorithms which allow the simplifying the mesh taking the color information into account [Hop99] [Gar98] [Roy05]. However, these algorithms do not provide a way to reconstruct the original mesh. To our knowledge, there is no progressive mesh coder allowing the encoding of color information.

2. DESCRIPTION OF BASE ALGORITHM

Our color compression scheme is based on the valence-driven progressive approach proposed by Alliez and Desbrun [All01a]. This algorithm uses the good statistical property of the native distribution of vertex valences for the mesh connectivity encoding. This approach iteratively decimates a set of vertices by combining decimation and cleansing conquests to get different levels of details (LOD). Decimation conquest consists in traversing the mesh patch by patch using a gate-based traversal; the front vertex of the current gate is removed only when its valence is below 7, in order to preserve compactly the vertex valence distribution. The hole left is then re-triangulated. The boundary edges of the actual patch are pushed into a FIFO list. The decimation conquest continues with the next available gate in the FIFO list, performing a breadth first traversal. Similarly, cleansing conquest removes only vertex of valence 3.

Fig.1 illustrates this mechanism: a regular input mesh (Fig.1.a) is simplified by decimation conquest (Fig.1.b). A set of independent vertices (red vertices) is removed and patches are re-triangulated. After performing cleansing conquest (Fig.1.c), vertices of valence 3 (blue vertices) are removed. We can see that as the input mesh is regular, the simplified mesh is also regular. Even for irregular meshes, this algorithm delivers better triangulation at coarse levels than the work of Cohen-Or et al. [Coh99]. During the compression stage, valences of removed vertices and additional null codes (in case of irregular mesh) are encoded for the connectivity.

For the geometry coding, Alliez and Desbrun first applied a global and uniform quantization to the coordinates of the mesh vertices. Then, they used both the barycentric prediction and the approximate Frenet coordinate frame, separating normal and tangential components to further optimize the bit rate. The base vectors of the local frame are built from the current gate (one of the boundary edges of the patch) and the approximated patch normal. The barycenter is obtained by averaging positions of neighboring vertices. The difference between the position of the vertex to be removed and the barycenter is then encoded in the local frame.

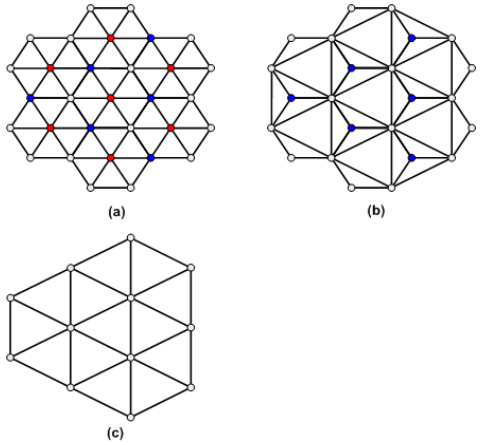


Figure 1. An example of decimation (b) and cleansing conquests (c) applied on a regular mesh (a).

Recently, Lee et al. [Lee09] proposed an improved geometric coder using a discrete bijection. They adopted the bijection method of Cartens et al. [Car99] and optimized the coding efficiency by providing an angle minimization. They also proposed a framework to improve the rate-distortion (R-D) trade-off by using adaptive quantization during the mesh simplification process.

In the following of this paper, we use the mesh traversal and the connectivity encoding techniques of [All01a] and the geometry coder of [Lee09].

3. COLOR COMPRESSION

The amount of color data associated to the mesh can be as large as or even larger than connectivity and geometry without an adaptive compression method. Therefore, a specific technique is required to reduce efficiently these data.

We propose in this section two methods which permit to encode the color data associated with mesh vertices, in a progressive manner.

Color space transform

Before to compress any color data, all colors expressed in the RGB space are transformed into the Lab space. The Lab space is the luminance-chrominance representation which describes more closely the human perception system. Moreover, this representation is more decorrelated than the RGB space. Thus, the Lab space is more appropriate to the

data compression. After this transformation, each color is represented using 8 bits for L, a and b color components as in the initial RGB space.

Prediction-based method

Since we consider the connectivity reduction of Alliez and Desbrun [All01a], the simplest method to predict the color value of the current vertex to decode is to use the average color of neighbors, like the prediction used for geometry encoding as illustrated in Fig. 2.

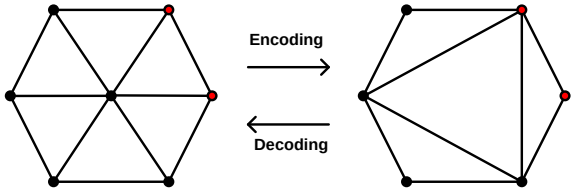


Figure 2. A vertex is removed (resp. inserted) during the encoding (resp. decoding) process. Its position is predicted from the averaged position of the neighboring vertices.

However, this prediction is not very efficient because the color data own a different behavior than geometry. In the case of quite regular meshes, the difference of positions (geometrical distance) between two vertices connected by an edge is relatively small, hence the barycentric prediction, explained in Section 2, can be performed efficiently. However, the color difference between two adjacent vertices can be very important, especially in the case of a vertex located in a color boundary, resulting that the averaging prediction is quite ineffective.

We can observe that the color value of a vertex is generally very close to at least one of its neighboring vertices' colors. Based on this observation, we propose a method which selects the proper color among the colors of the neighboring vertices so as to predict more efficiently. To perform this color selection, we first calculate the average values, L_{mean}, a_{mean} and b_{mean} of the neighbor colors. Then, for each component, we select the one which is the closest to the corresponding average component among the neighboring vertices' colors. The difference between the original and the selected color component values is then entropy coded to allow the decoder to reconstruct the exact color value. During the decompression process, after an insertion of new vertex, the corresponding color data is added to the vertex, allowing the progressive reconstruction.

Mapping table method

As each vertex color is represented using 24 bits, there exist 2^{24} possible colors. Yet, the human visual perception system cannot distinguish relatively small change of colors. Hence, we propose a method to reduce the bit rate needed for color encoding by reducing the number of colors to encode.

Our method first applies a clustering algorithm to the input mesh in order to reduce the number of possible colors without seriously affecting the visual distortion. Then, we use a mapping table method as in [Ahn06], based on the observation that this method is particularly useful when there is small number of colors. Fig.3 illustrates the diagram of our method in the case of the compression process.

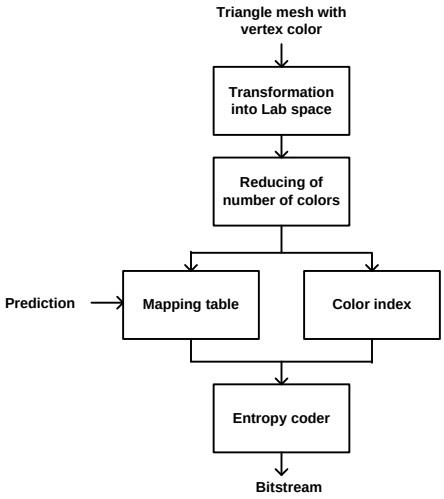


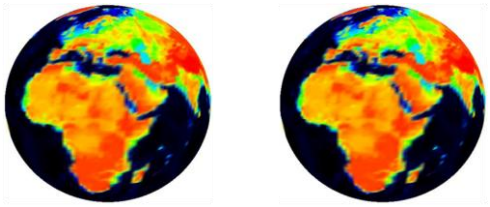
Figure 3. Diagram of the encoding process of our second algorithm.

The clustering method is widely used for 2D image compression [Sal98]. It consists in finding a set of representatives (Look Up Table) and in mapping each vertex color to its nearest representative. To generate a correct mapping table by minimizing the color distortion as much as possible, we use the well-known K-means clustering algorithm.

1. K initial seeds colors are selected from the mesh color data set.
2. K clusters are created by associating each color to the nearest seed.
3. The centroids of each cluster are used as new seeds and the new clusters are created.

The algorithm repeats step 2 and 3 until the all seeds are unchanged. Since the efficiency of the clustering algorithm depends on the initial condition of the seeds, we use as initial seeds the K more frequent colors of the input mesh in order to strengthen the approximation. After finding K representatives, each vertex color is replaced by its closest representative.

A result of this clustering algorithm is illustrated in Fig.4 with the Globe model containing initially 5030 colors. Although the number of possible colors is reduced to 256 colors, one can hardly distinguish the color distortion.



(a) Original model 5030 colors (b) Model represented with 256 colors

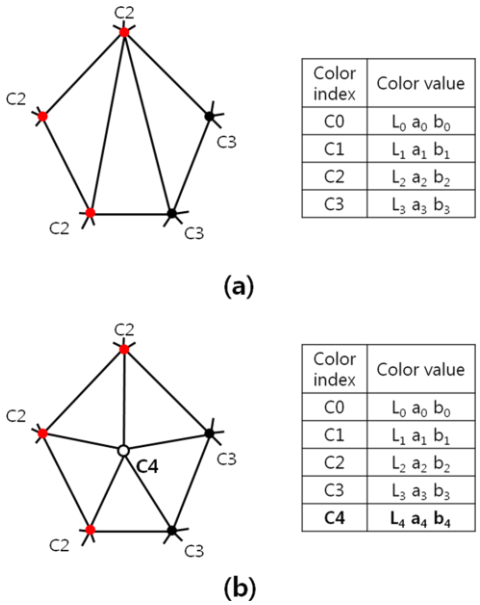
Figure 4. Color reduction based on clustering for the Globe model.

To encode the color data, we use the mapping table containing the final representatives obtained by the clustering algorithm. At the compression stage, when removing a vertex, the color index corresponding to its color in the mapping table is encoded.

To further enhance the rate-distortion performance and also to reduce additionally the coding cost, all color values contained in the mapping table are encoded in a progressive way. When the resolution level is augmented (when the mesh is refined to one higher level), the information of new colors are sent, enlarging the size of the mapping table. Fig. 5 illustrates an example of the progressive decoding of the mapping table. For a given resolution level, the mapping table contains 4 colors (C0 to C3). When a new vertex is inserted, and if the decoder identifies that the associated color is not present in the current mapping table then the new color value is added to it.

Furthermore, we try to reduce the coding cost needed for the encoding of the mapping table. In Ahn et al.'s work [Ahn06], they encode each color values in the mapping table using 24 bits. We reduce this coding cost by using our prediction-based method. During the compression process, we use our prediction method when removing each vertex. And we store only the difference between the original color value and the predicted color of the last encountered vertex for each color of the mapping table. So, during the mesh reconstruction, when a vertex is inserted and its color is revealed for the first time, we use information of the neighbors to acquire the correct color value of the corresponding color in the mapping table.

Even when the full resolution of the geometry has been reached, there still exist some differences of colors between the reconstructed color mesh and the original one, due to the color number reduction step (i.e. the clustering). However, depending on the needs, the original vertex colors can be restored, by encoding the difference of color between the initial color value of each vertex and its representative during the clustering phase. These differences are sent at the end of the decompression process.



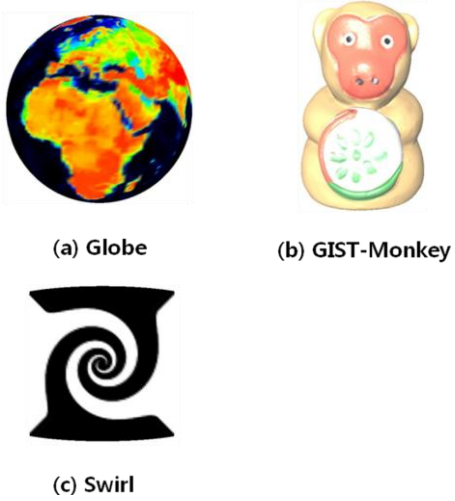
Color index	Color value
C0	L_0 a_0 b_0
C1	L_1 a_1 b_1
C2	L_2 a_2 b_2
C3	L_3 a_3 b_3

Color index	Color value
C0	L_0 a_0 b_0
C1	L_1 a_1 b_1
C2	L_2 a_2 b_2
C3	L_3 a_3 b_3
C4	**L_4 a_4 b_4**

Figure 5. An example of progressive decoding of the mapping table. Initial mapping table (a) is enlarged when a new color, C4, appears (b).

4. EXPERIMENTAL RESULTS

Fig. 6 shows the 3D models used in our experiments. Each coordinate of vertices of these models is quantized using 10 bits.



(a) Globe (b) GIST-Monkey (c) Swirl

Figure 6 : Models used for compression.

Lossless compression

Table 1 shows lossless compression results for the test models using our methods. The bit rates needed for compression of the color information and those of the mesh connectivity and the geometry (C+G) are given in bits-per-vertex (bpv). As most of the well-known state-of-the-art progressive algorithms do not consider color data, the efficiency of our prediction method is compared with the prediction scheme used in Yoon et al.'s work [Yoo07] and the averaging prediction. The method of Yoon et al. was originally applied in a single-rate way in their work. We have adapted their prediction method based on angle analysis for the mesh traversal technique of [All01a]. We can observe that the performance of these prediction schemes is similar for each model and better compression rates are obtained for the models containing large surface of smooth color variation, such as GIST-Monkey and Swirl models. For all test models, our method outperforms those of [Yoo07] and the averaging prediction method, especially for the Swirl model which contains many color boundary vertices and for those the color difference on the boundary is important.

Results of lossless compression of our mapping table method are also given. Different numbers of seeds, K, are used during the color number reduction step. We can see that the more the number of initial seeds increases, the more the coding rates decreases. This is because the cost of the original color restitution applied after reaching the finest geometry resolution level increases rapidly when the value of K becomes smaller. As a consequence, the result of the mapping table is better than our prediction method when the value of K is superior to 256.

Progressive compression

Fig. 7 illustrates some intermediates meshes with respective coding rates. All the rates presented in this figure include the amount of connectivity, geometry and color data. Our two methods produce intermediates results with a quite good visual quality both for the geometry and the color even for low bit rates (< 5 bpv).

In this figure, the GIST-Monkey model is used to compare the efficiency of our two methods: prediction method (d–f) and mapping table method (g–i). As expected, the mapping table method produces intermediate meshes of similar visual quality with less bit rates. Even though the number of colors has been severely reduced, from 6669 to 32, one can hardly sense the discrepancy comparing to the results of the prediction method.

5. CONCLUSION

In this paper, we have presented two methods for progressive encoding of colored meshes. To our knowledge the proposed methods are the first ones which consider the effective color coding in the field of 3D progressive compression. Our first algorithm based on the prediction is easily implementable and produces quite good results even for low bit rates. The second algorithm combining the mapping table with the clustering delivers intermediate meshes of almost equal visual quality with fewer bits, enhancing the rate-distortion trade-off.

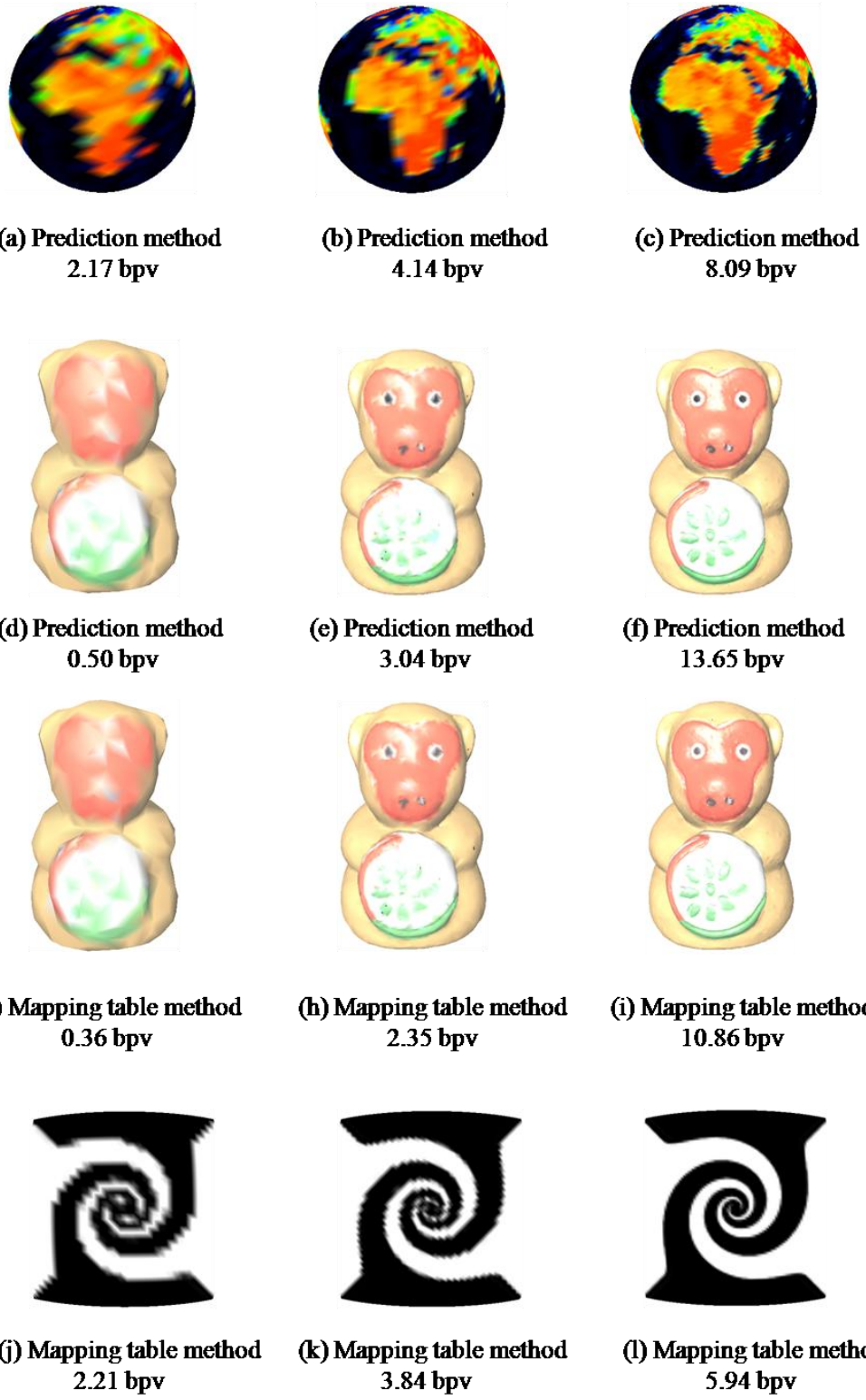
As future work, we will investigate a reliable metric permitting to measure the global distortion between two meshes taking mesh geometry and also color into account, in order to evaluate the rate-distortion performance.

ACKNOWLEDGMENTS

We would like to thank Hyun Soo Kim for sending us the color mesh models. This work has been supported by French National Research Agency (ANR) through COSINUS program (project COLLAVIZ n°ANR-08-COSI-003).

Models	# V	# Color	C + G	Prediction			Mapping table		
				Average	Yoon	Our	K = 64	K = 256	K = 1024
Globe	36866	5030	4.61	16.43	16.17	15.37	15.81	13.81	12.65
GIST-Monkey	50503	6669	13.5	6.49	6.49	5.95	8.52	8.33	7.23
Swirl	9216	138	4.12	9.97	10.16	6.62	3.04	-	-

Table 1. Compression rates of test models in bits-per-vertex.



(a) Prediction method 2.17 bpv
(b) Prediction method 4.14 bpv
(c) Prediction method 8.09 bpv
(d) Prediction method 0.50 bpv
(e) Prediction method 3.04 bpv
(f) Prediction method 13.65 bpv
(g) Mapping table method 0.36 bpv
(h) Mapping table method 2.35 bpv
(i) Mapping table method 10.86 bpv
(j) Mapping table method 2.21 bpv
(k) Mapping table method 3.84 bpv
(l) Mapping table method 5.94 bpv

Figure 7. Result of progressive decoding of the test models. The model Globe (a – c) and the model GIST-Monkey (d – f) are progressively reconstructed using our prediction method. Intermediates meshes of the models GIST-Monkey (g – i) and Swirl (j – l) are given by our mapping table method. For both models, the number of possible colors are reduced, using K = 32 seeds in the clustering step. The bit rates include the connectivity, the geometry and the color information.

REFERENCES

[Ahn06] J. Ahn, C. Kim, Y. Ho. Predictive compression of geometry, color and normal data of 3-D mesh models. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):291-299, 2006.

[All01a] P. Alliez and M. Desbrun. Progressive compression for lossless transmission of triangle meshes. In *ACM SIGGRAPH*, 198-205, 2001.

[All01b] P. Alliez and M. Desbrun. Valence-Driven connectivity encoding for 3D meshes. In *Eurographics*, 480-489, 2001.

[Baj99] C. L. Bajaj, V. Pascucci, and G. Zhuang. Single resolution compression of arbitrary triangular meshes with properties. In *IEEE Visualization*, 1999, 307-316.

[Car99] H.-G. Cartens, W.A. Deuber, W. Thumser, and E. Koppenrade. Geometrical bijections in discrete lattices. *Combinatorics, Probability and Computing*. 8:109-129, 1999.

[Coh99] D. Cohen-Or, D. Levin, and O.Remez. Progressive compression of arbitrary triangular meshes. In *IEEE Visualization Conference Proceedings*, 67-72, 1999.

[Dee95] M. Deering. Geometry compression. In ACM SIGGRAPH, 13-20, 1995.

[Gan02] P.-M. Gandoin and O. Devillers. Progressive lossless compression of arbitrary simplicial complexes. *ACM Transactions on Graphics*, 21(3):372-379, 2002.

[Gar98] M. Garland and P. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *IEEE Visualization*, 263-269, 1998.

[Gu02] X. Gu, S. Gortler, and H. Hoppe. Geometry Images. *ACM Transactions on Graphics*, 21(3):355-361, 2002.

[Gum98] S. Gumhold and W. Strasser. Real time compression of triangle mesh connectivity. In *ACM SIGGRAPH*, 133-140, 1998.

[Hop96] H. Hoppes. Progressive meshes. In *ACM SIGGRAPH*, 99-108, 1996.

[Hop99] H. Hoppes. New quadric metric for simplifying meshes with appearance attributes, In *IEEE Visualization*, 59-66, 1999.

[Kar02] Z. Karni, A. Bogomjakov, and C. Gotsman. Efficient compression and rendering of multi-resolution meshes. In *IEEE Visualization Conference Proceedings*, 347-354, 2002.

[Lee09] H. Lee, G. Lavoué, and F. Dupont. Adaptive coarse-to-fine quantization for optimizing rate-distortion of progressive mesh compression. In *VMV*, 73-81, 2009.

[Lee08] T. Lee, Y. Wang, and T. Chen. Animation key-frame extraction and simplification using deformable analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(4):478-486, 2008.

[Paj00] R. Pajarola and J. Rossignac. Compressed progressive meshes. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):79-93, 2000.

[Pen05] J. Peng and C.-C.J. Kuo. Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. In *ACM SIGGRAPH*, 609-616, 2005.

[Ros99] J. Rossignac. Edgebreaker : Connectivity compression for triangle meshes. *IEEE Transaction on Visualization and Computer Graphics*, 5(1):57-61, 1999.

[Roy05] M. Roy, S. Foutou, A. Koschan, F. Truchetet, and M. Abidi. Multiresolution analysis for meshes with appearance attributes, In *ICIP*, 816-819, 2005

[Sal98] D. Salomon. Data compression: The complete reference. Springer Verlag, 1998.

[Tau98a] G. Taubin, A. Guéziec, W. Horn, and F. Lazarus. Progressive forest split compression. In *ACM SIGGRAPH*, 123-132, 1998.

[Tau98b] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Transaction on Graphics*, 17(2):84-115, 1998.

[Tou98] C. Touma and C. Gotsman. Triangle mesh compression, In *Proceedings of Graphics Interface*, 26-34, 1998.

[Yao08] Z. Yao and T. Lee. Adaptive Geometry Image. IEEE Transactions on Visualization and Computer Graphics, 14(4):948-960, 2008.

[Yoo07] Y. Yoon, S. Kim, and Y. Ho. Color data coding for three-dimensional mesh models considering connectivity and geometry information. In *ICME*, 253-256, 2007.