Sequential Mesh Coding Using Wave Partitioning

Tae-Wan Kim[1], Kyoung Won Min[1], Byeong Ho Choi[1], and Yo-Sung Ho[2]

[1] Korea Electronics Technology Institute (KETI),
#68, Yatap-dong, Pundang-gu, Sungnam-Si, Kyunggi-do, 463-816, Korea
{kimtw, minkw, bhchoi}@keti.re.kr
[2] Gwangju Institute of Science and Technology (GIST),
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
hoyo@gist.ac.kr

Abstract. Recently, various three-dimensional(3-D) mesh coding schemes have been proposed to improve compression efficiency or error resilience. However, we need to consider both coding efficiency and error resilience when we send 3-D mesh data over bandwidth-limited transmission channels. In this paper, we propose a sequential mesh coding algorithm using the vertex pedigree based on the wave partitioning. After a 3-D mesh model is partitioned into several small processing blocks (SPB) using wave partitioning, we obtain vertices for each SPB along circumferences defined by outer edges of the attached triangles. Once all the vertices within each circumference are arranged into one line, we can encode the mesh model. Computer simulations show that the proposed algorithm provides both higher error resilience and improved coding efficiency.

1 Introduction

In recent days, three-dimensional (3-D) synthetic VRML models have been used in various applications, such as computer animation, computer vision, and studio graphic design. However, such mesh representation has a problem that 3-D objects with fine details have an excessive amount of data. Therefore, compression of mesh models is necessary to transmit them efficiently over a bandwidth-limited transmission channels.

Hoppe proposed a progressive mesh coding algorithm using mesh optimization to represent a 3-D model by a base mesh and vertex split variables; thus, the model can be shown in a progressive way using mesh simplification [1]. Taubin and Rossignac proposed a topological surgery method using vertex and triangle spanning trees to encode the connectivity and geometry data [2]. Touma and Gotsman improved coding efficiency of triangle mesh compression through traversal ordering using vertex degrees [3]. These schemes have focused on mesh compression and mesh simplification. If any bit errors occur during the transmission of 3-D model data, the reconstructed model can be severely damaged. Yan et al. suggested a robust coding scheme for 3-D graphic models using mesh segmentation and data partitioning, where the mesh model is partitioned and each partitioned mesh is encoded and transmitted in the packet to enhance error resilience. However, they did not consider coding efficiency seriously [4][5][6].

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part II, LNCS 3768, pp. 514 – 525, 2005.

In this paper, we propose a sequential mesh coding scheme using wave partitioning, aiming at both coding efficiency and error resilience [10][11]. After we partition a 3-D mesh model into several pieces, we define a mother vertex and its son vertices based on their topological relationship. We describe this relationship in detail in Chapter 3. Then, we encode the 3-D mesh model based on the index difference of each mother vertex of the current vertex and the next vertex with additional parameters, which is described in Chapter 4.

2 Wave Partitioning

In general, mesh partitioning and mesh segmentation techniques are used to divide the 3-D model into a set of small independent parts. Even though errors occur in one part, they will not affect the decoding of other parts in the model because the errors are limited to one part rather than the whole model. Therefore, when the channel error rate is high, mesh partitioning and mesh segmentation schemes are very effective. In the proposed algorithm, after the 3-D input model is partitioned, each partitioned unit is separated to several small processing block units. Encoding and decoding operations are performed on these small processing block units.

2.1 Basic Principle

The wave partitioning is simply based on the natural wave phenomenon that one drop of water is dropped on the water surface and spreads out, making circles in the lake.

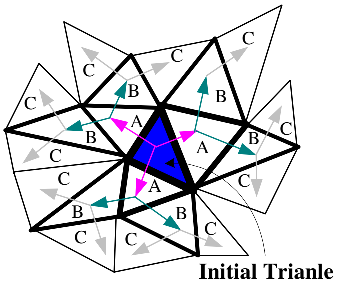


Fig. 1. SPB Formation Using Wave Partitioning

In the wave partitioning, an initial triangle is chosen and a number of triangles are attached to the initial triangle along the arrow direction. As shown in Fig. 1, triangles *A* which are sharing edges of the initial triangle, triangles *B* which are sharing edges of the triangles *A*, triangles *C* which are sharing edges of the triangles *B*, are attached to the initial triangle consecutively. Some attached triangles from the initial triangle consist of one partitioned part of the model.

This partitioned part of the model is called as SPB(Small Processing Block). In order to obtain a uniform size of SPB, we place two initial triangles on both ends of the model. After the whole model is divided into half, each part can be partitioned into several SPBs recursively. If we place one initial triangle on the model, the model is not partitioned into several pieces.

2.2 Modified Boundary Smoothing (MBS)

In order to reduce the boundary length and the processing complexity as much as possible, we apply a boundary smoothing algorithm [4]. As illustrated in Fig. 2, holes in one separate SPB can be filled up by projected triangles from its adjacent SPB. If there are *N* vertices on the SPB boundary and the model is partitioned into *A SPB* and *B SPB* in Fig. 2, choose one of these *N* vertices (v_i) in *SPB A* and v_k on the boundary except v_{i-1} and v_{i+1}. Check whether there exist the edge (v_i,v_k) and the path ($v_i \rightarrow v_{i-1} \rightarrow v_k$) in *SPB A*. If there are (v_i,v_k) and ($v_i \rightarrow v_{i-1} \rightarrow v_k$) in *SPB A*, we regard that there exists a hole in *SPB B*. By removing the path ($v_i \rightarrow v_{i-1} \rightarrow v_k$) from the boundary in *SPB A* and adding the edge (v_i,v_k) into the boundary in *SPB B*, holes in *SPB B* are filled up by triangles in *SPB A*.

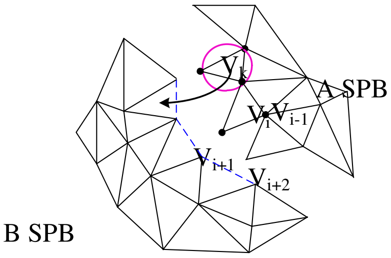
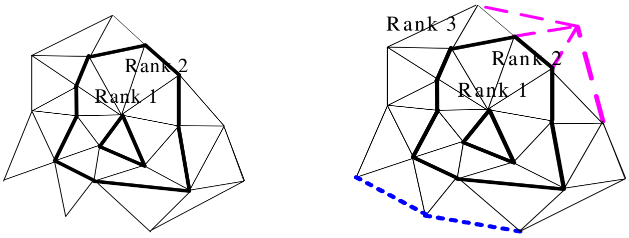


Fig. 2. Modified Boundary Smoothing Scheme

When we partition a mesh model into several SPBs by using the wave partitioning, the partitioned SPB can be described, as shown in Fig. 3(a). The bold lines in Fig. 3(a)(b) are defined as the Rank in the SPB.



(a) Before MBS (b) After MBS

Fig. 3. Modified Boundary Smoothing Scheme

If we apply the boundary smoothing scheme [4] to the partitioned SPB, small dotted triangles are attached to the partitioned SPB, as shown in Fig. 3(b). However, the outer line does not construct the rank. We add the long dotted triangles to the partitioned SPB which are already applied by the boundary smoothing scheme, as shown in Fig. 3(b). Then, *Rank 3* is constructed. This scheme is defined as modified boundary scheme (MBS). Figure 4 shows the procedure in which one model is partitioned into two SPBs and two SPBs are partitioned into four SPBs again. MBS followed by wave partitioning is also applied.

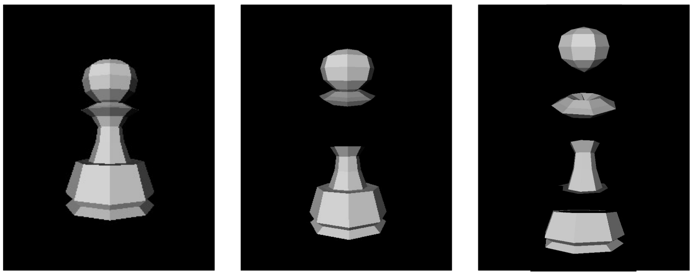
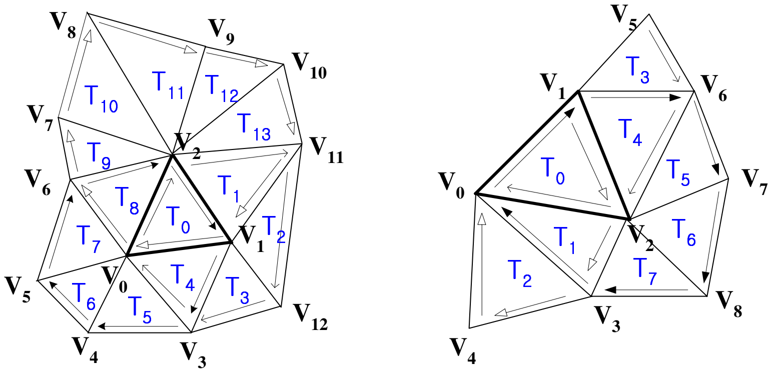


Fig. 4. Object Applied with MBS & Wave Partitioning Scheme

3 Two Types of SPBs

SPBs are classified into circular or semi-circular types based on the location of the initial triangle. In the case that the initial triangle is located on the inner space of the mesh, these SPBs are called as a circular SPB. On the contrary, if the initial triangle is placed on the boundary of the mesh, these SPBs are called as a semi-circular SPB.



(a) Circular SPB (b) Semi-circular SPB

Fig. 5. Two Types of SPB

Figure 5(a) shows an example of the circular SPB. The closed thick line in Fig. 5(a) that is composed of v_0, v_1 and v_2 is defined as *Rank 1*. The path composed of vertices $v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{12}$ surrounding *Rank 1* is defined as *Rank 2* consecutively. Vertices surrounding a particular vertex, except for those vertices that are already included in the inner rank, are defined as son vertices(*Sv*) of the particular vertex, which is called as the mother vertex(*Mv*) of its surrounding son vertices. In Fig. 5(a), v_0 is surrounded with vertices $v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_2$. Because v_1 and v_2 are already included in the *Rank 1*, v_0 is a *Mv* of v_3~v_6 except v_1 and v_2. v_1 is surrounded with vertices $v_0 \rightarrow v_2 \rightarrow v_{11} \rightarrow v_{12} \rightarrow v_3$. As v_1 and v_2 are already included in the *Rank 1* and v_3 is a *Sv* of v_0, v_1 is a *Mv* of v_{11}~v_{12}. In the same way, v_2 becomes a *Mv* of v_7~v_{10}.

Figure 5(b) shows an example of the semi-circular SPB. Compared to a circular SPB, the left part of path $v_4 \rightarrow v_1 \rightarrow v_2 \rightarrow v_5$ is not included in the semi-circular SPB and the rank does not consist of the closed loop. However, the principle of making ranks and relations between a *Mv* and a *Sv* is the same as that of the circular SPB. Therefore, v_0 is a *Mv* of v_3~v_4, v_1 is a *Mv* of v_5~v_6 and v_2 is a *Mv* of v_7~v_8.

3.1 Circular SPB

Figure 6 shows a partitioned circular SPB by wave partitioning and MBS. The thick lines in Fig. 6 indicate ranks in a SPB. Vertices of the initial triangle, v_0~v_2, consist of *Rank 1* and the path composed of $v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{10} \rightarrow v_{11}$ becomes *Rank 2*. When we arrange all the vertices included in each rank in Fig. 6 into a row on the plane, we can form a map of the vertices, as shown in Fig. 7.

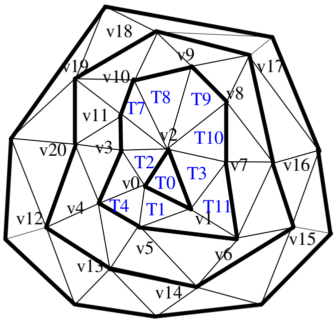


Fig. 6. A Partitioned Circular SPB

In Fig. 7, each horizontal line lists the vertices of the same rank and the mother-son relationship in two adjacent ranks is indicated by thick lines. Following the vertex ordering in ranks, we classify each vertex into one of four types: *Cv* is the current vertex, *Nv* is the next vertex, Mv_1 is the mother vertex of *Cv*, and Mv_2 is the mother vertex of *Nv*. Whereas v_0 is a *Mv* of v_3~v_5, v_1 is a *Mv* of v_6~v_7. If v_5 is a *Cv*, v_6 is a *Nv*, v_0 is a Mv_1 and v_1 is a Mv_2. Since each rank forms a closed loop in the circular SPB, the first vertex in each rank reappears at the last position of the rank in Fig. 7. In order to represent the topological information of the vertices, we define the index of each vertex by its distance from the first vertex in the same rank and express the mother-son relationship by encoding the indices of their mother vertices differentially.

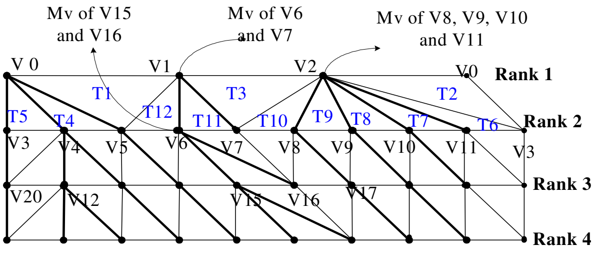


Fig. 7. Vertex Arrangement of the Circular SPB

In Fig. 7, v_6 in *Rank 2* is the mother vertex of v_{16}. Since v_6 is the third vertex from v_3, its index is 3. Similarly, v_8 in *Rank 2* is the mother vertex of v_{17} and its index is 5. Thus, we can represent the topological relationship of v_{16} and v_{17} by the index difference of their mother vertices, v_6 and v_8, i.e., 5 – 3 = 2.

3.2 Semi-circular SPB

Figure 8 shows an example of the semi-circular SPB. As in the circular SPB, we can define the rank: *Rank 1* includes v_0~v_2 and *Rank 2* includes v_3~v_8. In Fig. 8, we indicate all ranks in the semi-circular SPB by thick lines. As shown in Fig. 8, each rank, except for the first rank, forms the semi-circular shape since there is no connecting edge between v_4 and $v_{5,}$ or between v_{11} and v_{12}.

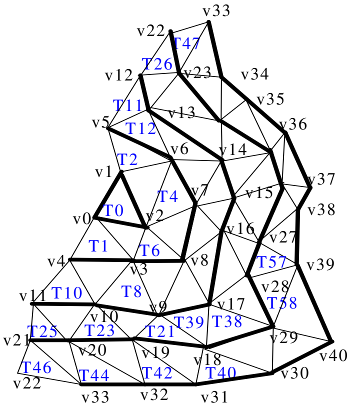


Fig. 8. A Partitioned Semi-Circular SPB

When we arrange all the vertices included in each rank in Fig. 8 into a row on the plane, we can form the map of the vertices, as shown in Fig. 9.

Since there is no connecting edge between v_4 and v_5 or between v_{11} and v_{12} in Fig. 8, we do not have a triangle containing v_4 and v_5 or v_{11} and v_{12} in Fig. 9. To make *Rank 2*, *Sv*'s of v_0 are found at first. Then, v_3 becomes the first vertex of *Rank 2* and shown in the last position in *Rank 2* again in Fig. 9.

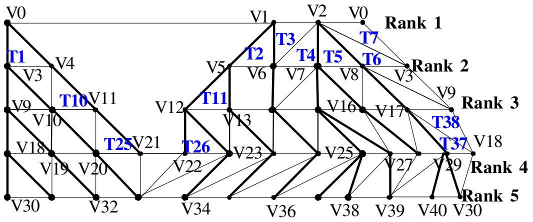


Fig. 9. Vertex Arrangement of a Semi-circular SPB

4 Sequential Encoding and Decoding

In the proposed algorithm, a mesh model is encoded and decoded in a SPB unit independently. Therefore, the model is partitioned into several SPBs and we obtain the map of the vertices, as shown in Fig. 7 and Fig. 9. Then, the encoding processes of Block A and Block B in Fig. 10 are applied to topological information and geometry information, respectively. In the case of topological information, as shown in Section 3.1, index differences which indicate the position from the first vertex in each rank, are acquired and encoded with the numbers of vertex in each rank. Both topological information and geometry information are coded in a rank unit to enhance error resilience. These processes are continued repeatedly until all SPBs will be encoded. Specially, we add the decoder and the comparator at the end of the encoder to transmit topological information that is not coded by our proposed algorithm. However, this data is very rare in most 3-D mesh models.

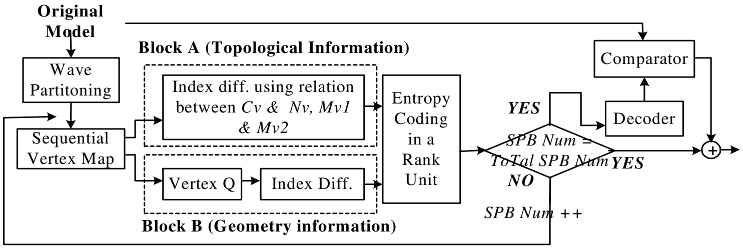

Original Model
Wave Partitoning
Sequential Vertex Map
Block A (Topological Information)
Index diff. using relation between Cv & Nv, Mv1 & Mv2
Vertex Q
Index Diff.
Block B (Geometry information)
Entropy Coding in a Rank Unit
YES
SPB Num = ToTal SPB Num
NO
YES
SPB Num ++
Comparator
Decoder
+

Fig. 10. Flow of the Encoding Algorithm

4.1 Topological Information

For the topological information, we transmit the numbers of vertices in each rank and $|Mv_1 - Mv_2|$ for each vertex at the encoder. The order of encoding is from left to right in a rank unit. Based on the value of $|Mv_1 - Mv_2|$, the proposed algorithm is classified into 4 types.

For the case of $|Mv_1 - Mv_2| = 0$, the triangle (Cv, Nv, Mv_1) is reconstructed at the decoder. In Fig. 7, when Cv is v_3, Nv is v_4, Mv_1 is v_0 and Mv_2 is v_0, $|Mv_1 - Mv_2| = |0 - 0| = 0$ and triangle (v_3, v_4, v_0) is reconstructed.

In the case of $|Mv_1 - Mv_2| = 1$, two triangles (Cv, Nv, Mv_2) and (Mv_1, Mv_2, Cv) are at the decoder. In Fig. 7, when Cv is v_7, Nv is v_8, Mv_1 is v_1 and Mv_2 is v_2, $|Mv_1 - Mv_2| = |1 - 2| = 1$. Triangles ($v_7, v_2, v_1$) and ($v_7, v_8, v_2$) are reconstructed.

In the case of $|Mv_1 - Mv_2| = 2$, we can consider 28 patterns, as shown in Fig. 11.

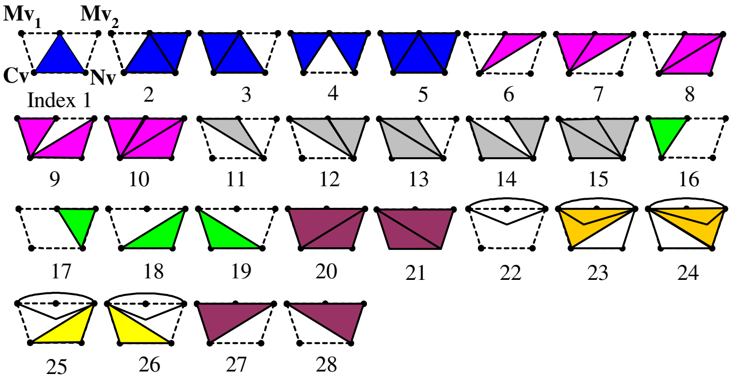


Fig. 11. Twenty-eight Patterns and Its Indices when $|Mv_1 - Mv_2| = 2$

At the decoder, one of 28 patterns is decoded based on the received index. So, in addition to the value of $|Mv_1 - Mv_2|$, 1 byte, which is composed of 5 index bits and 3 reserved bits, for a pattern index is also added to the bit-stream. Triangles that are composed of dotted edges are not real triangles. The cone shape in the 22nd ~ 26th patterns represents the triangle that is composed of Mv_1, Mv_2 and one vertex between Mv_1 and Mv_2. In Fig. 7, when Cv is v_{16}, Nv is v_{17}, Mv_1 is v_6 and Mv_2 is v_8, $|Mv_1 - Mv_2| = |3 - 5| = 2$. This case corresponds to the 10th pattern in Fig. 11. Index 10 is transmitted at the encoder and the 10th pattern is reconstructed at the decoder.

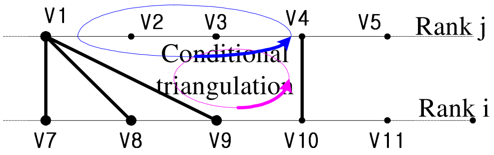


Fig. 12. Conditional Triangulation

If $|Mv_1 - Mv_2| > 2$, triangles are made conditionally, as shown in Fig. 12. According to the locations of vertices, triangles can be classified into three types. Table 1 indicates the types and indices for the conditional triangulation of Fig. 12. In Fig. 12, when v_9 in the *Rank i* is regarded as Cv, v_{10} is regarded as Nv, v_1 as Mv_1 and v_4 as Mv_2. Then $|Mv_1 – Mv_2|$ equals to 3 and triangles indicated in Table 1 will be made. Type I indicates triangles composed of v_9 or v_{10} in *Rank i* and two of v_1 ~ v_4 *Rank j*. Type II indicates triangles composed of v_9 and v_{10} in *Rank i* and one of v_1 ~ v_4 in *Rank j*. Type III indicates

Table 1. Vertex Types and Indices for Conditional Triangulation

	Indices of each triangle		
Type I	(v_9,v_1,v_2): 0,	(v_9,v_1,v_3): 1,	(v_9,v_1,v_4): 2,
	(v_9,v_2,v_3): 3,	(v_9,v_2,v_4): 4,	(v_9,v_3,v_4): 5,
	(v_{10},v_4,v_3): 6	(v_{10},v_4,v_2): 7,	(v_{10},v_4,v_1): 8,
	(v_{10},v_3,v_2): 9,	(v_{10},v_3,v_1): 10,	(v_{10},v_2,v_1): 11
Type II	(v_9,v_{10},v_1) : 12,	(v_9,v_{10},v_3) : 14,	
	(v_9,v_{10},v_2) : 13,	(v_9,v_{10},v_4) : 15	
Type III	(v_1,v_2,v_3) : 16,	(v_2,v_3,v_4) : 18,	
	(v_1,v_2,v_4) : 17,	(v_1,v_3,v_4) : 19	

triangles composed of 3 of v_1 ~ v_4 in *Rank j*. In this case, more than one triangle can be made. Therefore, one byte which is composed of 5 bits for indices in Table 1 and three bits for the number of triangles is encoded for each vertex in the encoder.

4.2 Geometry Information

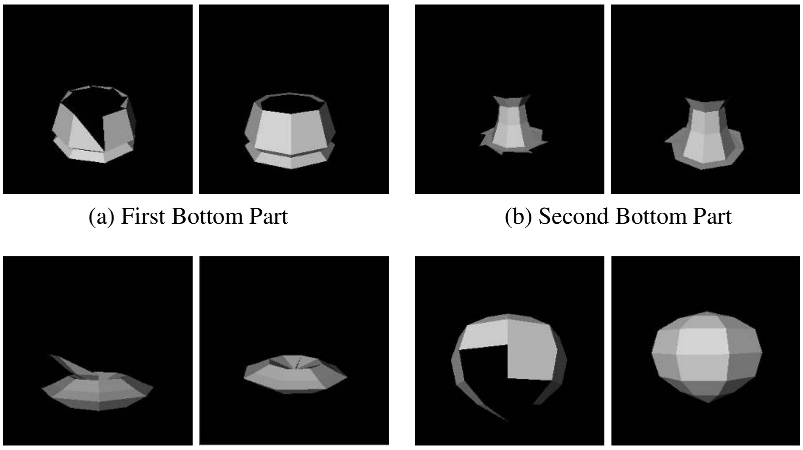
For geometry information, the difference between Cv and Nv in the same rank quantized and entropy coded. Then, their indices are transmitted. As the geometry information is encoded in the same order as the topological information, models can be decoded using the values of $|Mv_1 - Mv_2|$ for the topological information and geometry information.

5 Experimental Results

Fig. 13 shows the result of MBS on the PAWN model. In Fig. 13 (a)(b)(c)(d), left images to which MBS is not applied show rough contours which cannot construct a rank. On the contrary, right images in Fig. 13(a)(b)(c)(d) show the result of wave partitioning and MBS scheme. If MBS followed by wave partitioning is not applied, the probability for the value of $|Mv_1 - Mv_2|$ to be more than two becomes higher. Then, additional indices must be added and coding efficiency can be lower. Besides, in some cases, the amount of residual data which is acquired by comparing the comparator with the decoder in Fig. 10 can be increased.

In order to evaluate the performance of the proposed scheme, we run computer simulations on several VRML mesh models and compare the results with those of the MPEG-4 SNHC standard which is based on the topological surgery.

Figure 14 demonstrates the test models that are decoded sequentially by the rank unit. As we put one initial seed triangle on the test models, test models are not partitioned into two or three pieces though we scan triangles to be encoded in the way of wave partitioning. While the topological information in each model is decoded losslessly, the geometrical information has been encoded by a 256-level quantizer and the arithmetic encoder. The coding efficiency is summarized in Table 2. We have improved coding efficiency for the topological information by about 1bit/vertex relative to the MPEG-4 SNHC standard. For geometrical information, even if coding efficiency depends on the mesh model, we get lower coding efficiency than that of the MPEG-4 SNHC algorithm by about less than 1 bit/vertex because we consider only one adjacent vertex to encode the current vertex in each rank.



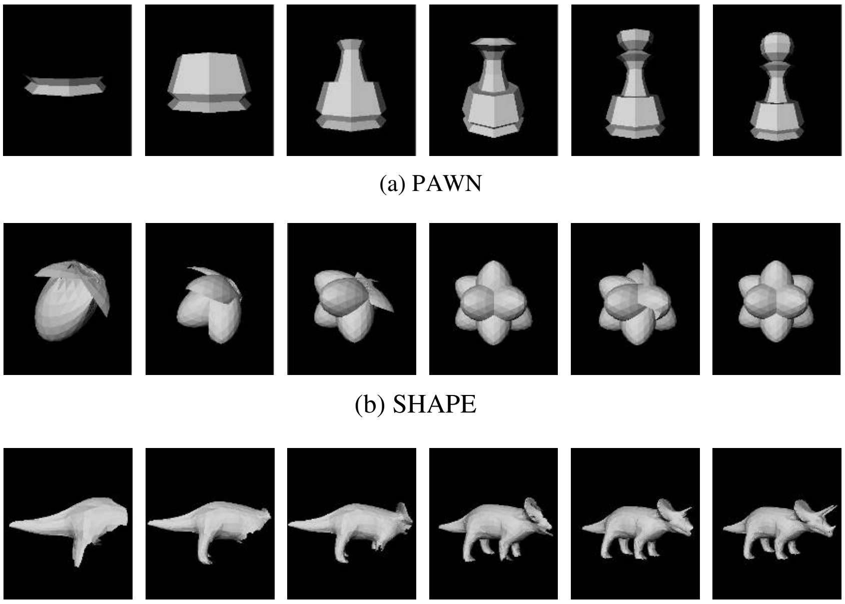
(a) First Bottom Part

(b) Second Bottom Part

(c) Second Top Part

(d) First Top Part

Fig. 13. Partitioned Pawn Model Applied with MBS



(a) PAWN

(b) SHAPE

(c) TRICERATOPS

Fig. 14. Sequential Decoding Results

Table 2. Coding Efficiency of Topological Information

	Model (Bytes)	Coding Efficiency (bits/vertex)			
		Proposed Algorithm		MPEG-4 SNHC	
		Topology	Geometry	Topology	Geometry
SHAPE	30,720	0.95	15.1	2.2	14.3
TRICERATOPS	33,960	3.31	11.2	4.3	10.4
BEETHOVEN	30,168	3.32	14.3	4.8	15

6 Conclusions

In this paper, we have proposed an efficient error resilient coding algorithm for 3-D mesh models based on wave partitioning by exploiting the topological and geometrical information. Since all the processing is executed rank by rank independently, the proposed algorithm is error resilient. We have also improved coding efficiency over the MPEG-4 SNHC coding scheme by about 1bit/vertex for the topological information. We have obtained reasonably good reconstructed 3-D models at 0.2~0.3 bit/vertex.

Acknowledgements

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

References

1. Hoppe, H.: Progressive Meshes. Computer Graphics, Proc. of SIGGRAPH (1993) 99-26
2. Taubin, G., Rossignac, J.: Geometry Compression through Topological Surgery. ACM Trans. Graphics, Vol. 17 (1998)
3. Touma, C., Gotsman, C.: Triangle Mesh Compression. Graphics Interface Conference Proc. (1998) 26-34
4. Yan, Z., Kumar, S., Li, J., Kuo, C.-C.J.: Robust Coding of 3D Graphic Models Mesh Segmentation and Data Partitioning. Proc. IEEE Int. Conference Image Proc., Oct. (1999)
5. Yan, Z., Kumar, S., Kuo, C.-C.J.: Mesh Segmentation Schemes for Error Resilient Coding of 3-D Graphic Models. Proc. IEEE Int. Trans. Circuits and Systems for Video Technology, Vol. 15, No. 1, Jan. (2005)
6. Yan, Z., Kumar, S., Kuo, C.-C.J.: Error-Resilient Coding of 3-D Graphic Models via Adaptive Mesh Segmentation. Proc. IEEE Int. Trans. Circuits and Systems for Video Technology, Vol. 11, No. 7, Jan. (2001)
7. Deering, M.: Geometric Compression. Computer Graphics Proc. (1995) 13-20
8. Taubin, G., Horn, W., Lazarus, F., Rossignac, J.: Geometric Coding and VRML. Proc. IEEE, Vol. 86 (1988)

9. Schroeder, W.J., Zarge, J.A.: Decimation of Triangle Meshes, Computer Graphics. Proc. of SIGGRAPH (1992) 65-70
10. Witten, I., Neal, Cleary, J.: Arithmetic Coding for Data Compression. Commun. ACM, Vol. 30, June (1987) 520-540
11. Bajaj, C.L., Pascucci, V., Zhuang, G.: Progressive Compression and Transmission of Arbitrary Triangular Meshes. Visualization '99, Proceedings, Oct. (1999) 307-537