MESH-BASED DEPTH CODING FOR 3D VIDEO USING HIERARCHICAL DECOMPOSITION OF DEPTH MAPS

Sung-Yeol Kim and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong Buk-gu, 500-712, Gwangju, Korea

ABSTRACT

In this paper, we present a new coding scheme for depth maps using a hierarchical decomposition. After we decompose a depth map into three disjoint images and a layer descriptor according to the region of edges, we merge the disjoint images of each depth map into an image. Then, the merged images and the layer descriptor are coded by H.264/AVC. Unlike previous mesh-based depth coding methods, we compress the irregular depth information using a conventional 2D video coder. Experimental results show that our scheme improves compression efficiency of mesh-based depth coding.

Index Terms— Mesh-based depth coding, hierarchical decomposition, 3D video, H.264/AVC

1. INTRODUCTION

As immersive multimedia services are expected to be available anytime and anywhere, we are very interested in the 3D video as high-quality visual media. Recently, depth image-based rendering (DIBR) [1] has been issued to represent a 3D video efficiently. DIBR uses a color image and the corresponding depth map to generate a 3D view, instead of stereoscopic images. Especially, we generate 3D scenes with DIBR in real time by employing a mesh structure [2].

The mesh-based 3D video is a collection of 3D scenes represented by color images and feature points extracted from depth maps. Feature points are depth pixels in a depth map that influence critically on the shape of 3D scenes. For rendering the mesh-based 3D video, we create a 3D surface with the feature points frame by frame, and then cover the surface with the color image using a texture mapping. The advantage of the mesh-based representation is the high rendering speed that allows us to enjoy 3D scenes in real time. However, it is hard to compress the feature point data of the mesh-based 3D video due to their irregularity.

One solution to remove the irregularity is that we regard all depth information in a depth map as feature points [3]. If so, we can code depth maps with a 2D video coding scheme, such as MPEG-2 or H.264/AVC. However, since this approach generates a huge number of triangles, it is almost impossible to render consecutive 3D scenes in real time. Another solution is a 3D warping technique [4]. Although high-performance works in the field of 3D warping have been developed, we still need reasonable hole-filling algorithms and more stable warping systems.

Grewatsch et al. [5] presented a mesh-based coding scheme for feature point data. They compress the 3D depth information using the MPEG-4 3DMC coder. *Chai et al*. [6] also introduced a depth map coding scheme using an adaptation of the triangular mesh generation. For real-time rendering, they matched the tree traversal information into the mesh rendering order. However, they should have maintained a complicated tree structure frame by frame, and needed a new coder to compress the mesh-based 3D video. *Morvan et al*. [7] proposed another scheme for depth coding, which employed quad-tree decomposition and plane approximation. They converted depth maps into several mode information and related depth data irregularly.

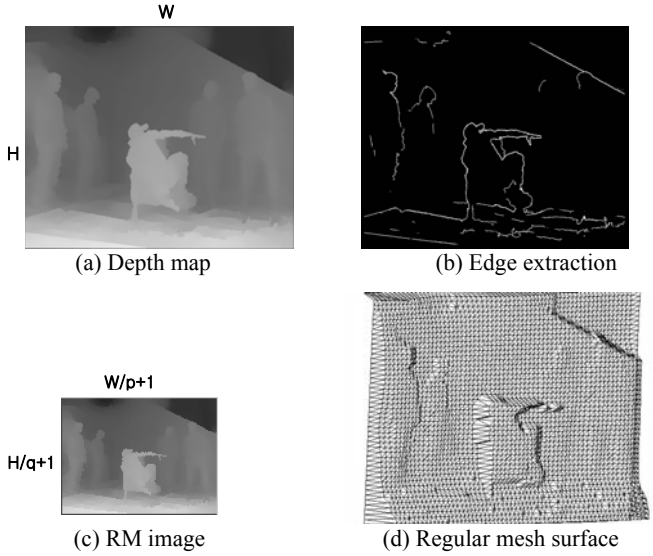
The main problem of previous works was that special 3D coders were needed to compress irregular feature point data. Naturally, the complexity of the coding system increased significantly because new coders have been added in the current video coding system. In this paper, we try to maintain the depth information of the mesh-based 3D video regularly. Therefore, we can use conventional 2D video coders directly to compress the depth information.

2. HIERARCHICAL DECOMPOSITION

In a hierarchical decomposition [8], depth maps are decomposed into four kinds of special images: regular mesh (RM), edge-region (ER), no-edge-region (NER), and number-of-layer (NOL) images. RM, ER, and NER images, which include feature point data, are disjointed one another. NOL image are used to manage the three disjoint images.

In the decomposition, we first extract edges by applying the *Sobel* filter into a depth map vertically and horizontally. Then, we divide the region of the depth map according to the size of the grid cell. The grid cell is the unit of the decomposition, and it is similar with a macroblock in 2D

video coding. If the size of the grid cell is $p \times q$, an RM image is generated by downsampling the depth map with the horizontal sampling rate p and the vertical sampling rate q. Each frame of the 3D video is represented by a set of lattice feature points. When the resolution of the depth map is $W \times H$, the resolution of the RM image is $(W/p+1) \times (H/q+1)$. Figure 1 shows the generation of an RM image.

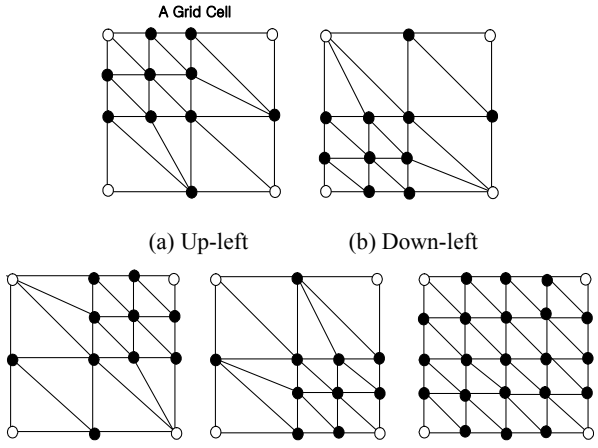


(a) Depth map (b) Edge extraction

(c) RM image (d) Regular mesh surface

Fig. 1. Generation of RM image

When the grid cell includes edges, we assign the depth information in the grid cell into an image, named by an ER image. The ER image is generated by four quad-tree modes and a full modeling mode. If the region of edges occupies more than a half of a grid cell, a full modeling mode is used. Otherwise, one of quad-tree modes is selected according to the location of edges in the grid cell.

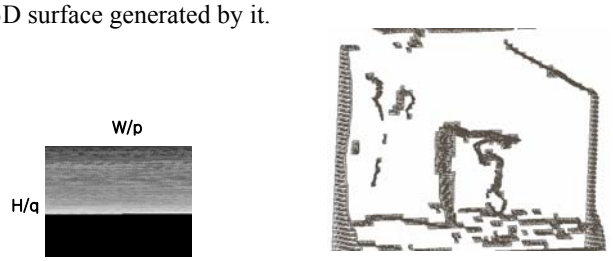


(a) Up-left (b) Down-left

(c) Up-right (d) Down-right (e) Full Modeling

Fig. 2. Modes of ER images

As shown in Fig. 2, ER images employ up-left, up-right, down-left, down-right quad-tree modes, and a full modeling mode. Dark feature points are assigned into the ER image along the raster scanning order. We need 10 feature points in the quad-tree mode, as shown in Fig. 2. On the other hand, we need 21 depth data to describe the grid cell by a full modeling mode. Figure 3 shows an ER image and the 3D surface generated by it.



(a) ER image (b) 3D surface of edge regions

Fig. 3. Generation of ER images

For the no edge region, we choose feature points in the grid cell according to the influence on the 3D surface. A maximum distance algorithm is employed to find out influent feature points. The most influent feature points are gathered into an image to generate the 1st NER image. Likewise, the 2nd influent feature points are also gathered into another image to generate the 2nd NER image. Since the 3D surface on the no edge region is smooth, we can describe the surfaces efficiently with one or two feature points.

Unlike RM and ER images, we should consider the x- and y-coordinate data to indicate the feature point location as well as depth values. In order to omit the coordinate data, we define four representatives to assign specific feature points. When we divide the grid cell into four sub-grid cells, the center positions of the sub-grid cells become positions of representatives. The feature point is mapped into the closest representative. Figure 4 shows an NER image and the final 3D surface generated by RM, ER and NER images.

(a) NER image (b) Final 3D surface

Fig. 4. Generation of NER images

In order to manage the structure of hierarchical decomposition of depth maps, we employ NOL images. Pixel intensities in the NOL image contain the information of the number of feature points and the mode information of ER images. When the maximum number of NER images is c, pixel intensities from $(c \times 4+1)$ to $(c \times 4+4)$ in the NOL image indicate four quad-tree modes. The pixel intensity for a full modeling mode is $(c \times 4+5)$. Naturally, pixel intensities from 0 to $(c \times 4)$ indicate the number of feature points in no edge region and the location of representatives. The NOL information is gathered into an image to generate NOL images.

3. COMPRESSION OF DEPTH INFORMATION

Figure 5 shows the proposed system to compress depth information of the mesh-based 3D video. First, we generate RM, ER, and NER images from depth maps using hierarchical decomposition. Then, we merge the decomposed images into one image. Finally, the H.264/AVC coder is used to compress the merged images. NOL images are coded by entropy coders directly.

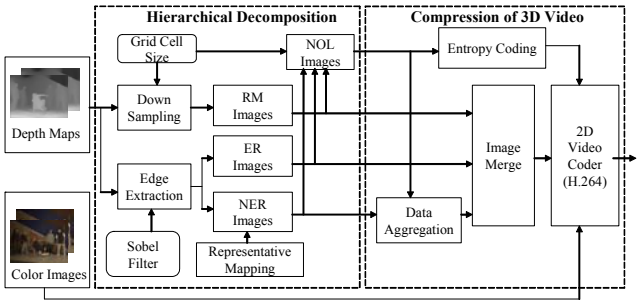


Fig. 5. Proposed depth coding system

As shown in Fig. 6, since there are lots of empty pixels in NER images, we aggregate scattered pixels along the vertical direction. Scattered pixels cause low coding efficiency by generating tremendous residual errors. Aggregated images can be recovered at the decoder completely because NOL images include the location of feature points.

(a) Before data aggregation (b) After data aggregation

Fig. 6. Data aggregation

After hierarchical decomposition of depth maps, RM, ER, and NER images are merged into one image sequence. When the size of the grid cell and the resolution of depth maps are $p \times q$ and $W \times H$, respectively, the resolution of the RM image is $(W/p+1)\times(H/q+1)$, and the resolution of ER and NER images is $(W/p \times H/q)$. In order to set the size of decomposed images to be equal, we remove the last vertical and horizontal lines of RM images. The removed information is restored at the decoder side by an interpolation technique. As a result, the resolution of merged images is $W/p\times(n(\text{ER})+n(\text{NER})+1)\times H/q$, where n(ER) and n(NER) indicate the numbers of ER and NER images, respectively.

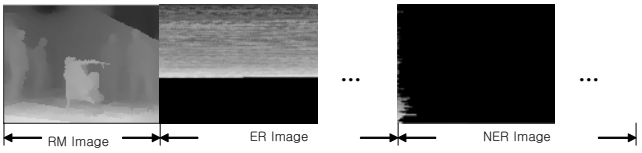
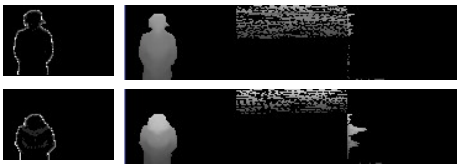


Fig. 7. Image merging

4. EXPERIMENTAL RESULTS AND ANALYSIS

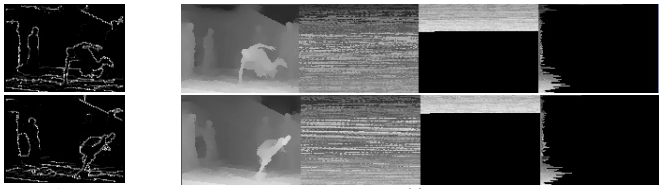
We have tested the performance of the proposed method with two tested sequences: 'Homeshopping' [8] and 'Breakdancers' [9]. While Homeshopping has 100 frames with 720×480 pixels, Breakdancers has 100 frames with 1024×768 pixels. We defined the size of the grid cell as 8×8. In order to code decomposed images by H.264/AVC, we have used the JM 9.6 reference software.

For Homeshopping, we generated an RM image, an ER image, and an NER image for each frame. The resolution of merged images of Homeshopping is 270×60. For Breakdancers, we used an RM image, two ER images, and an NER image for each frame. The resolution of merged images for Breakdancers is 512×96. Figure 8 shows the result of the decomposition of the 10th and 20th depth maps.



NOL Merged images

(a) Homeshopping



NOL Merged images

(b) Breakdancers

Fig. 8. Hierarchical decomposition

We compared our scheme to the Grewatsch's approach, which used MPEG-4 3DMC to code the depth data. Besides, we compared our scheme to the H.264/AVC approach, which regarded all depth information of depth maps as feature points. As shown in Fig. 9, our scheme was 0.8 ~ 1.2 dB better than the MPEG-4 3DMC approach in the terms of coding efficiency. Connectivity and texture coordinate data for 3D scenes were not needed to code in the proposed scheme, because we maintained data regularity. However, the H.264/AVC approach was better than our scheme, because the spatial and temporal correlations of the original depth maps were much higher than the decomposed images, although the resolution of decomposed images was much smaller than the original depth maps.

We compared the rendering times of our scheme and the H.264/AVC approach. In order to create 3D surfaces with feature points in the H.264/AVC approach, we employed a full modeling technique, which is similar to the full modeling mode of ER images. As shown in Table 1, our proposed scheme had higher rendering speed by 20 times than the H.264/AVC approach. We can reduce the number of trian-

gles efficiently while maintaining data regularity. In addition, we can reconstruct 3D scenes successfully using mesh triangulation. Figure 10 shows the rendering result of 3D scenes generated by our proposed scheme.

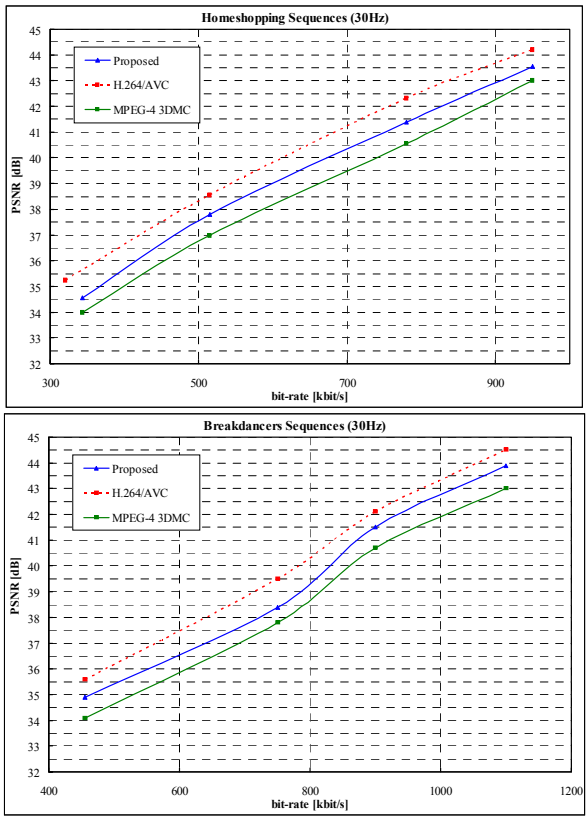


Fig. 9. Results of depth map coding

20th frame 30th frame

(a) Homeshopping

10th frame 18th frame

(b) Breakdancers

Fig. 10. Reconstruction of a mesh-based 3D video

Table 1: Comparison of rendering time

Test Data	H.264/AVC Approach		Proposed scheme	
	Number of triangles	Frame rates	Number of triangles	Frame rates
H.S.	172,800	0.65 fps	5,670	12.82 fps
B.D.	402,432	0.51 fps.	24,705	10.21 fps

5. CONCLUSTIONS

In this paper, we proposed a new scheme to compress depth information of a mesh-based 3D video. Since previous methods did not choose feature points regularly, they could not use conventional 2D video coders to compress the feature point data. Since we maintained the data regularity with hierarchical decomposition of depth maps, we were able to encode the decomposed images using H.264/AVC. Experimental results showed that our scheme improved coding efficiency of mesh-based depth coding in comparison to previous works by 0.8dB ~ 1.0dB. Furthermore, we could render dynamic 3D scenes in real time.

ACKNOWLEDGMENT

This work was supported in part by MIC through RBRC at GIST, and in part by MOE through the BK21 project

REFERENCES

[1] A. Ignatenko and A. Konushin, “A framework for depth image-based modeling and rendering,” *Proc. of Graphicon*, pp. 169-172, 2003.

[2] G. Taubin and J. Rossignac, “Geometric compression through topological surgery,” *ACM Trans. on Graphics*, vol. 17, pp. 84-115, 1998.

[3] C. Fehn, K. Schuur, P. Kauff, and A. Smolic, “Coding results for EE4 in MPEG 3DAV,” ISO/IEC/JTC1/SC29/WG11, M9561, 2003.

[4] J. Shade, S.J. Gorler, and R. Szeliski, “Layered depth image,” *SIGGRAPH*, pp. 291-298, 1998.

[5] S. Grewatsch and E. Muller, "Fast mesh-based coding of depth map sequences for efficient 3D video reproduction using OpenGL," *Visualization, Imaging, and Image Processing*, 480-66, 2005.

[6] B. Chai, S. Sethuraman, H. Sawhey, and P. Hatract, “Depth map compression for real-time view-based rendering,” *Pattern Recognition Letter*, vol. 25, pp. 755-766, 2004.

[7] Y. Morvan, D. Farin, and P.H.N. de With, “Novel coding technique for depth images using quadtree decomposition and plane approximation,” *Proc. of Visual Communication and Image Processing*, pp. 1187-1194. 2005.

[8] S.Y. Kim, S.B. Lee, and Y.S. Ho, “Three-dimensional natural video system based on layered representation of depth maps,” *IEEE Trans. on Consumer Electronics*, vol. 52, no. 3, pp. 1035-1042, 2006.

[9] Interactive Visual Media Group at Microsoft Research, http://research.microsoft.com/vision/InteractiveVisualMediaGroup/3DVideoDownload/.