3DTV SYSTEM USING DEPTH IMAGE-BASED VIDEO IN THE MPEG-4 MULTIMEDIA FRAMEWORK

[1]Sung-Yeol Kim, [1]Jongeun Cha, [2]Seung-Hyun Lee, [1]Jeha Ryu, and [1]Yo-Sung Ho

[1]Gwangju Institute of Science and Technology, Korea
[2]Kwangwoon University, Korea

ABSTRACT

In this paper, we present a 3DTV system using a new depth image-based representation (DIBR). After obtaining depth images from multi-view cameras or a depth-range camera, we decompose the depth image into three disjoint layer images and a layer descriptor image. Then, we combine the decomposed images with color images to generate a new representation of dynamic 3D scenes, called as a 3D depth video. The 3D depth video is compressed by a H.264/AVC coder, and streamed to clients over IP networks in the MPEG-4 multimedia framework. The proposed 3DTV system enables not only enjoying a high-quality 3D video in real time, but also experiencing various user-friendly interactions such as free viewpoint changing, composition of computer graphics, and even haptic display.

Index Terms**—**3D depth video, Depth image-based representation, MPEG-4 multimedia framework, 3DTV

1. INTRODUCTION

Three-dimensional (3D) TV [1] is considered as the next-generation broadcasting supporting high-quality audio-visual services and various user-friendly interactions, such as view changing. In the ATTEST project [2], it gave us an opportunity to realize a 3DTV system. Recently, the development of the commercial European 3D broadcasting system has been in progress with a new project, named by 3DTV [3]. Meanwhile, multi-view video coding (MVC) has been a big issue in the MPEG standardization [4].

Although a multi-view video, acquired from multi-view camera systems, can satisfy the requirement of 3DTV in the aspect of free viewpoint viewing, it still has some limitations to support immersive interactions, such as intermediate view reconstruction and scene composition with computer graphic models. Recently, depth image-based representation (DIBR) [5] has introduced as one of main technologies for 3DTV. DIBR represents consecutive 3D scenes by color images and corresponding depth images.

In order to represent and render a 3D scene with DIBR techniques, we can employ mesh structures. In the mesh-based representation [6], we extract feature points from depth images and generate 3D scenes using mesh triangulation. Color images are then used to cover the 3D geometry surfaces. The main advantage of the mesh-based representation is a high rendering speed to reconstruct 3D dynamic scenes in real time. However, it is hard to compress its data due to irregularity. In other words, we need a special 3D coder to compress connectivity data to recognize positions of feature points, which are not constant frame by frame. Therefore, a 3DTV system will be complicated by adding a special coder.

A solution to remove the data irregularity in mesh-based representation is to render 3D scenes with all depth information in depth images. However, the approach has a problem to render dynamic 3D scenes in real time because the number of triangles to render increases exponentially. Other solution is the image-based rendering (IBR) technique that uses 3D warping to generate new views. Although good performance works have been developed in the field of 3D warping, we still need a reasonable hole-filling algorithm.

In fact, most previous works related to 3DTV techniques focused on developing efficient representations and compression schemes for 3D data, not system issues such as adding interactive functionality and multiplexing various multimedia data. In this paper, we present a 3DTV system under the MPEG-4 multimedia framework supporting various user-friendly interactions. Moreover, we introduce a new image format, called as *3D depth video*, of mesh-based representation in DIBR. The 3D depth video lets us render consecutive 3D scenes in real time, and maintains the data regularity to be compatible with the existing video system.

2. SYSTEM ARCHITECTURE

Figure 1 shows the overall system architecture of the proposed 3DTV system. First, depth and color images are obtained from a depth-range camera or multi-view cameras. After decomposing a depth image into three disjoint layer images and a layer descriptor image using information decomposition, we combine color images with the decomposed images to generate a 3D depth video. Then, the 3D depth video is compressed by a H.264/AVC coder and multiplexed with other data, such as audios and computer graphic models, under the MPEG-4 multimedia framework.

At a client side, we extract the 3D depth video by a demultiplexer and H.264/AVC decoder. After decomposing the 3D depth video into a color image, three disjoint layer images and a layer descriptor image for each frame, we generate a 3D geometry surface with four decomposed images. Finally, we render 3D scenes with color images utilizing a texture mapping technique.

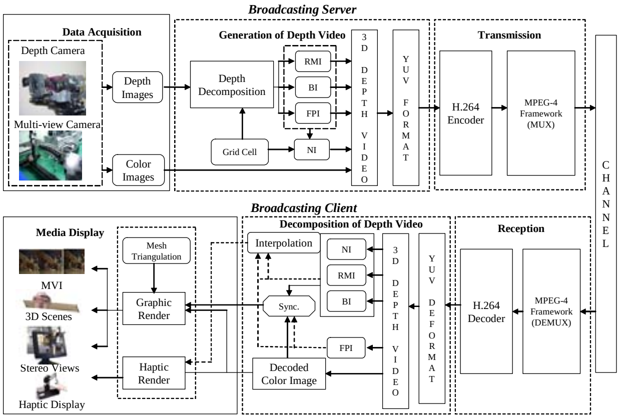


Fig. 1. Overall system architecture

3. GENERATION OF 3D DEPTH VIDEO

3.1. Information decomposition of depth images

Information decomposition of depth images [7] converts depth images into three disjoint layer images: regular mesh images (RMIs), boundary images (BIs), and feature point images (FPIs). We also employ number-of-layer images (NIs) to manage the three disjoint images. As shown in Fig. 2(a), a grid cell is defined as a unit of the information decomposition of depth images. When the size of a grid cell is *p*×*q*, we obtain a RMI by downsampling a depth image with a horizontal sampling rate *p* and a vertical sampling rate *q*, as shown in Fig. 2(b). Thurs, we extract 4 depth data from each grid to generate a RMI.

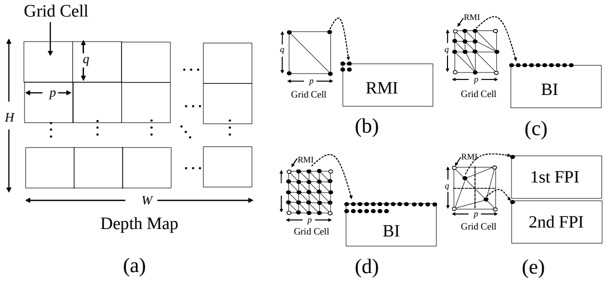


Fig. 2. Information decomposition of depth images

A BI includes depth data at the region of edges in a depth image. We describe the grid cell on the region of edges with a quad-tree mode and a full modeling mode. Figure 2(c) and Figure 2(d) show a quad-tree mode and a full modeling mode, respectively. When the region of edges occupies more than a half of a grid cell, we use a full modeling mode. Otherwise, we use a quad-tree mode. In a quad-tree or a full modeling mode, we need 10 or 21 depth data, respectively. The depth data are gathered into BIs. A FPI has depth data at the region of no edges. As shown in Fig. 2(e), we extract feature points from a grid cell on the region of no edges, and gather them into FPIs. In addition, we generate a NI to manage the number of feature points for a grid cell on the region of no edges and the mode information for a grid cell on the region of edges.

3.2. 3D depth video

After information decomposition of depth images, we merge the RMIs, BIs, FPIs, and NIs with corresponding color images to generate a 3D depth video. Before merging the images, we change the gray-level space of RMIs, BIs, NIs, and FPIs into RGB color space by repeating their intensities three times. Also, we multiply the data of NIs by a weight value *k* to prevent from losing their information during data compression. When the maximum number of FPIs is *b*, the multiplier *k* is defined by [255/(b+5)], where [·] represents the flooring operation. Figure 3 shows a frame of 3D depth video. The upper side of the frame shows a color image, and the lower side shows a NI, a RMI, BIs, and FPIs.

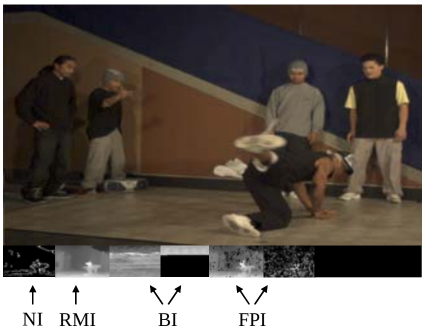


Fig. 3. A frame of 3D depth video

The 3D depth video has advantages in system simplicity and easy data processing. When we transmit depth and color images separately, we need two data processors to deal with their streams, and we should solve the synchronization problem between them. Also, since we convert 3D information in depth images into 2D information, we do not need any special 3D data processor, such as a MPEG-4 3D mesh coder (3DMC).

4. MPEG-4 NODE SPECIFICATION

The 3D depth video contains appearance information from the color image and geometrical information of the scene from decomposed images obtained by depth images. Therefore, it is reasonable to define the 3D depth video node in the *Shape* node. In MPEG-4 BIFS, a *DepthImage* node is already defined. However, it is not included in the

Shape node, but added in the *Transform* node. The *DepthImage* node itself describes its pose and textures. In this paper, a new *DepthMovie* node that is able to be stored in the geometry field is designed by following the concept of the *Shape* node by shown in Fig. 4.

```
DepthMovie
{
        field    SFVec2f     fieldOfView     0.785398 0.785398
        field    SFFloat     nearPlane       10
        field    SFFloat     farPlane        100
        field    SFBool      orthographic    TRUE
}
```

Fig. 4. DepthMovie node specification

The fields of the *DepthMovie* node are the same as the fields of the *DepthImage* node that includes the camera parameters. Here, the *DepthMovie* node does not contain the data of a 3D depth video. Rather, in MPEG-4 BIFS, the 2D video is described with the *MovieTexture* node and it is stored on the texture field of the *Appearance* node. Hence, *MovieTexture* node can be used for a 3D depth video.

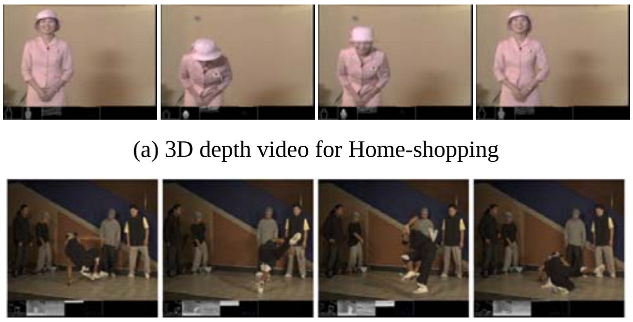
5. TRANSMISSION OF 3D DEPTH VIDEO

We use a H.264/AVC coder to compress a 3D depth video. Before converting a RGB stream into a YUV stream, we add horizontal lines with a default value, 255, into the 3D depth video to match the horizontal resolution of the 3D video to multiples of 16. The compressed 3D depth video and the BIFS stream, which includes the description of a 3D depth video using the *DepthMovie* node, are multiplexed into a *MP4* file that is designed to contain the media information of an MPEG-4 presentation. The MP4 file can be played from local hard disk and over IP networks. A Darwin Streaming Server (DSS) [8], which supports to stream MPEG-4 content over the Internet in real time or on demand, is used as a streaming server. Viewers can enjoy the contents in the context of the video-on-demand concept.

6. EXPERIMENTAL RESULTS

We have evaluated the performance of our proposed system with two test sequences, Home-shopping [9] and Breakdancers [10, 11]. Home-shopping has 100 frames with 720×480 resolution, whereas Breakdancers has 100 frames with 1024×768 resolution. Home-shopping was captured by a depth-range camera, *ZCam*^{TM}, whereas Breakdancers was obtained by multi-view cameras using a stereo matching.

In the experiment, we set the size of grid cell as 8×8. Figure 5 shows the 1st, 20th, 40th, and 60th frame of 3D depth videos of two test sequences. In Home-shopping, we made a NI, a RMI, a BI and a FPI for a frame, and the resolution of the 3D depth video was 720×544. In Breakdancers, we made a NI, a RMI, two BIs, and a FPI for a frame, and the resolution of the 3D depth video was 1024×880.

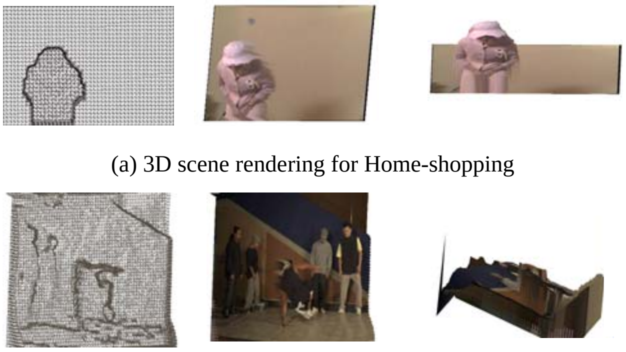


(a) 3D depth video for Home-shopping

(b) 3D depth video for Breakdancers

Fig. 5. Generation of 3D depth video

Figure 6 shows the result of 3D scenes generated by the 3D depth video. We reconstructed 3D scenes successfully using mesh triangulation. As shown in wire frame of 3D scenes, the region of edges was more detailed than others because our scheme separated the feature points to render according the amount of influence on the 3D surface.



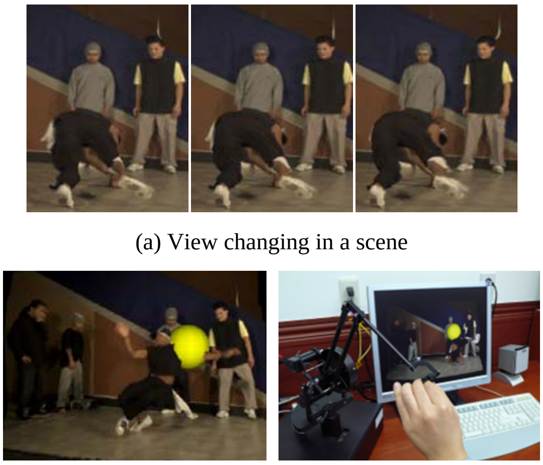
(a) 3D scene rendering for Home-shopping

(b) 3D scene rendering for Breakdancers

Fig. 6. Rendering results from 3D depth video

Figure 7 shows the possible interactions when the 3D depth video is employed in a 3DTV system. As shown in Fig. 7(a), viewers could change the viewpoint freely. Since the depth video was rendered with a mesh structure, viewers could enjoy smooth viewpoint change contrary to the multi-view video. Viewers could also enjoy the rich scene composed of a computer graphic and the 3D depth video. Since the depth video has 3D information, the sphere mesh model, which was represented with *IndexedFaceSet* node in BIFS, could be inserted as shown in Fig. 7(b). Finally, viewers could touch the shape of the 3D depth video wearing a haptic device by applying a haptic rendering algorithm [12] as shown in Fig. 7(c).

In the sense of maintaining the data regularity, we compared the rendering time between our scheme and a 3D full modeling. A 3D full modeling uses all depth information in a depth image to render a 3D scene. As shown in Fig. 8, the rendering speed of our scheme had about 20 times higher than the 3D full modeling.



(a) View changing in a scene

(b) 3D composition with CG (c) Haptic interaction

Fig. 7. Possible interactions with 3D depth video

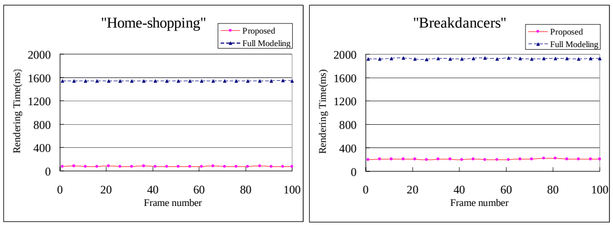
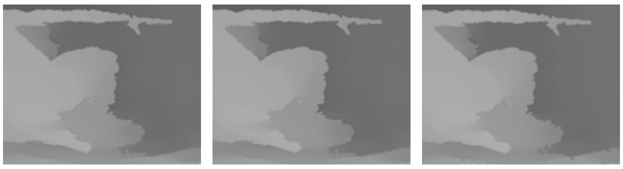


Fig. 8. Rendering time comparison



(a) Original (b) Grid cell: 4×4 (c) Grid cell: 8×8

Fig. 9. Visual quality of depth images

Table 1. Depth quality estimation

Test Seq.	PSNR (dB), when QP in a H.264/AVC coder is 30			
	Original	Grid Cell Size		
		4×4	8×8	16×16
H.S.	45.02	44.29	39.28	34.33
B.D.	42.47	42.35	38.45	30.27

In addition, we evaluated the quality of depth images generated by decomposed images. Figure 9 shows the original and the interpolated depth image generated by decomposed images when the size of grid cell is 4×4 and 8×8. Table 1 shows variations of the visual quality interpolated depth images according to the size of grid cell. The more the size of grid cell was, the more visual degradations the interpolated depth images had. Hence, we should choose the size of grid cell reasonably according to the network situations and target applications.

7. CONCLUSIONS

In this paper, we introduced a new 3DTV system using a 3D depth video. Experimental results show that the proposed system can render 3D dynamic scenes in real time without serious visual quality degradations. Furthermore, the proposed system can support interactive functionalities for view changing, composition of computer graphics, and even haptic interaction under the MPEG-4 multimedia framework. We hope that the proposed 3DTV system presents new directions and improves the quality of various multimedia services in 3-D broadcasting.

ACKNOWLEDGMENT

This work was supported by the MIC, Korea, under the ITRC support program supervised by the IITA and in part by MOE through the BK21 project.

REFERENCES

[1] C. Fehn, R. de la Barre, and S. Pastoor, “Interactive 3DTV Concepts and Key Technologies,” *Proceedings of the IEEE*, vol. 94, no. 3, pp. 524-538, 2006.

[2] A. Redert, M. op de Beeck, C. Fehn, W. IJsselsteijn, M. Pollefeys, L. van Gool, E. Ofek, I. Sexton, and P. Surman, “ATTEST – Advanced Three-Dimensional Television System Technologies,” *Proc. of International Symposium on 3D Data Processing*, pp. 313–319, 2002.

[3] 3DTV consortium, http://3DTV.zcu.cz/.

[4] ISO/IEC JTC1/SC29/WG11 N7779, Subjective test results for the CfP on multi-view video coding, 2006.

[5] A. Ignatenko and A. Konushin, “A Framework for Depth Image-Based Modeling and Rendering,” *Proc. of Graphicon*, pp. 169-172, 2003.

[6] S. Grewatsch and E. Muller, “Fast Mesh-Based Coding of Depth Map Sequences for Efficient 3D Video Reproduction Using OpenGL,” *Proc. of International Conference on Visualization*, Imaging and Image Processing, 2005.

[7] 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.

[8] Apple Computer, Inc. Darwin Streaming Server, http://developer.apple.com/opensource/server/streaming/, 2006

[9] J. Cha, S.M. Kim, S.Y. Kim. S.U. Yoon, I. Oakley, J. Ryu, K.H. Lee, W. Woo, and Y.S. Ho, “Client System for Realistic Broadcasting: A First Prototype,” *Lecture Notes in Computer Science*, vol. 3768, pp. 176-186, 2005.

[10] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-Quality Video View Interpolation Using a Layered Representation,” *SIGGRAPH*, pp. 600-608, 2004

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

[12] J. Cha, S.Y. Kim, Y.S. Ho, and J. Ryu, “3D Video Player System with Haptic Interaction Based on Depth Image-Based Representation,” *IEEE Trans. on Consumer Electronics*, vol. 52, no. 2, pp. 477-484, 2006.