Multi-view Depth Video Coding using Depth View Synthesis

Sang-Tae Na, Kwan-Jung Oh, Cheon Lee, and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong Buk-gu, Gwangju, 500-712, Korea
{stna, kjoh81, leecheon, and hoyo}@gist.ac.kr

Abstract— **Depth information indicates the distance of an object in the three dimensional (3D) scene from the camera view-point, typically represented by eight bits. Since the depth map is useful in various multimedia applications, such as three dimensional television (3DTV) and free-viewpoint television (FTV), we need to acquire a single or multi-view depth maps and process them effectively. In this paper, we propose a new coding scheme for multi-view depth video data using depth view synthesis. We first apply a 3D warping method to synthesize a virtual depth image for the current view using the multi-view depth information. We also propose a hole filling method to compensate for the holes generated during the depth map synthesis process. Finally, we utilize the synthesized depth map for the current view as an additional reference frame in encoding the current depth map. Experimental results show that the proposed algorithm achieves approximately 0.69 dB of PSNR gain on average, compared to JMVM 1.0.**

I. INTRODUCTION

As multimedia processing technologies advance, demands for interactive contents are also increasing dramatically. We can reproduce natural and realistic three-dimensional (3D) scenes by combining the depth data with the conventional two-dimensional (2D) scene for three-dimensional television (3DTV). In free-viewpoint television (FTV), we can generate virtual views that provide us unrestricted spatio-temporal navigation using the depth data [1]. Those systems need both single or multi-view depth video data, and texture video data. As more cameras are used, the amount of multi-view depth video data is also increasing enormously. Therefore, we need to develop efficient coding methods for such applications.

The multi-view video coding has been studied earlier than multi-view depth video coding. In 2001, new standardization activities have been launched by Ad-Hoc Group (AHG) on 3-D audio and visual (3DAV). In October 2004, it has been reported that multi-view video coding algorithms give much better results than simulcast coding with H.264/AVC [2]. As a result, the joint multi-view video model (JMVM) based on the joint scalable video model (JSVM) [3] has been proposed, and the software has been selected as the reference software for multi-view video coding (MVC) in October 2006. In April 2007, the multi-view video plus depth (MVD) format has been presented for advanced future video systems, such as 3DTV and FTV [4].

Generally speaking, multi-view depth video data can be encoded by utilizing their spatial and temporal correlation. Since multi-view depth video consists of multiple depth data for the same scene, they have high correlations both in the spatial and temporal domains [5].

In this paper, we propose an efficient method for multi-view depth video coding. After we synthesize a virtual depth map for the current view using a 3D warping method, we fill the holes generated during the warping process by copying the neighboring pixels or using other virtual view. Then, we use the virtual depth map for the current view as an additional reference frame for encoding of the current depth map.

II. PREVIOUS METHOD FOR MULTI-VIEW DEPTH VIDEO CODING BASED ON MVC

Previously, the multi-view depth video data are encoded independently at each viewpoint by H.264/AVC, as shown in Fig. 1. This independent coding structure with hierarchical B picture coding for temporal prediction is referred to as simulcast coding. Figure 1 illustrates a sample coding structure when three cameras are used and the length of the group of pictures (GOP) is eight. In Fig. 1, S_n denotes the individual view sequence and T_n denotes the consecutive time point.

The multi-view depth video data consists of multiple depth maps of the same scene in the same manner as the multi-view color video data. Thus, the multi-view depth video data can be compressed using the multi-view video coding scheme where we can exploit inter-view redundancy as well as temporal redundancy to achieve better coding efficiency. Even though hierarchical B picture coding has shown to provide high compression efficiency in the temporal domain, its coding efficiency is somewhat limited in the multi-view depth video coding.

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

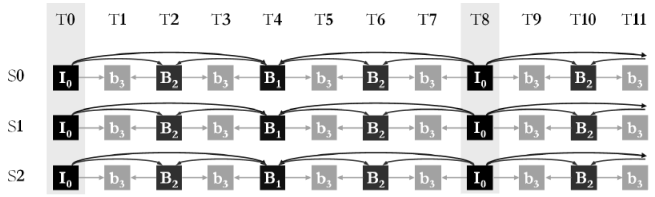

Figure 1. Simulcast coding structure with hierarchical B pictures

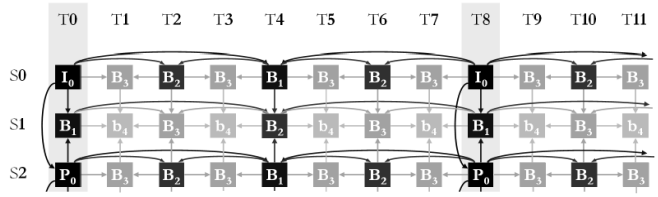

Figure 2. Multi-view coding structure with hierarchical B pictures

However, prediction from inter-view reference frames can provide some potential to improve coding efficiency for multi-view depth video data as well as for multi-view video data. In order to exploit all statistical dependencies within a multi-view data set, inter-view prediction has to be combined with temporal prediction. Figure 2 illustrates a new coding structure which allows inter-view prediction [5].

III. Multi-view Depth Video Coding using Depth View Synthesis

In order to compress the multi-view depth video data, we develop a new method that utilizes the characteristics of the depth video and multiple color videos of the same scene.

As mentioned earlier in the previous section, we can apply the multi-view video coding scheme on the multi-view depth video because they have similar characteristics. Thus, we develop our scheme based on JMVM that is currently the reference software for the MVC standardization work.

In order to exploit the characteristics of the depth video, we adopt an idea from the view synthesis prediction method [6]. When we encode the current depth view, we synthesize a virtual depth map for the current view by applying a 3D warping technique. Then, we utilize the virtual depth map for the current view as an additional reference frame. In the proposed scheme, we do not need any other side information to synthesize the virtual depth maps except camera parameters because the depth map itself can be used as both the reference video and the depth data. For texture video, we need the corresponding depth data as well as camera parameters as the side information.

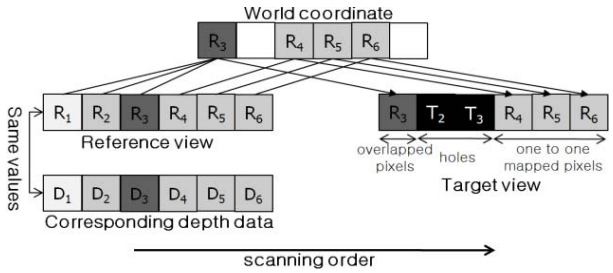

Figure 3. Pixel position miss matching

We perform the 3D warping process for view synthesis. Firstly, we apply the pinhole camera model to project the pixel position in the reference depth image into the 3D world coordinates by

$$P_{WC} = {R_{ref}}^{-1} \cdot {A_{ref}}^{-1} \cdot P_{ref} \cdot D - {R_{ref}}^{-1} t_{ref} \quad (1)$$

where R, A and t denote the rotation matrix, intrinsic matrix and translation vector for the reference or target depth image, respectively, and these values are called as camera parameters. D is the depth value, and P is the pixel position on the 3D world coordinates or the reference image. After the above projection, the 3D pixel position is projected again into the position in the desired target image by

$$P_{target} = A_{target} \cdot R_{target} \cdot (P_{WC} + {R_{target}}^{-1} \cdot t_{target}) \quad (2)$$

Then, we can obtain the right pixel position in the target image with respect to the pixel position in the reference image. Finally, we share the pixel value between the pixel position in the reference image and the projected pixel position in the target image.

During this process, none or more than one pixel position in the reference image can be projected into one position in the target image due to the occlusion and disoccluion regions, as shown in Fig. 3. Occlusion regions are defined as areas which exist at the reference image, but invisible at the target image. Disocclusion regions are areas which cannot be seen at the reference image, but exist at the target image.

Since the pixel positions in disocclusion regions of the target image are not projected by any pixel position from the reference image, pixel values remain as the initial value and the regions are named as holes. To fill the holes efficiently, we consider two cases: (ⅰ) when only one directional reference view is available, such as predicted view (P-view), and (ⅱ) when both directional reference views are available, such as bi-directional view (B-view).

For the former case, we synthesize a virtual P-view using reconstructed intra view (I-view) or P-view. To fill the holes in the 3D warped view, we refer pixel values from the background areas because holes are usually generated in some parts of the background areas hidden by the foreground areas in the reference image, but are visible in the target image. Consequently, we use the neighboring pixel values of holes only from the background areas.

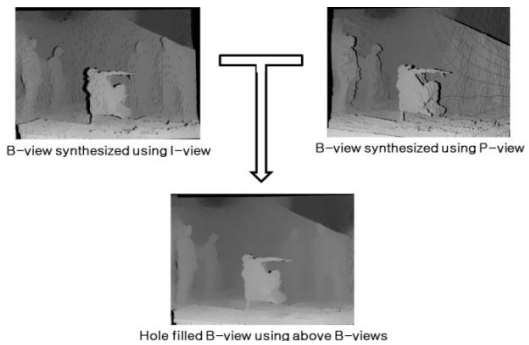

Figure 4. Hole filling using both directioin of reference views

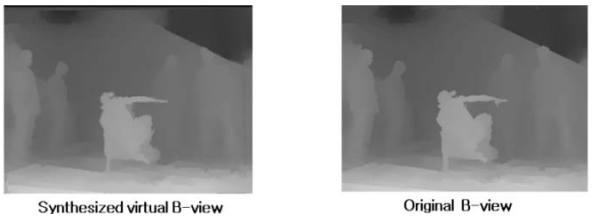

Figure 5. Comparing virtual view to original view

Therefore, we need to distinguish background areas from foreground areas. By scanning the virtual depth image, we can easily separate those areas. If the pixel values in the left side of the holes are bigger than the values in the right side, holes are located in the right side of the foreground areas; otherwise, holes are located in the left side of the foreground areas. Since the depth images only include the distance information, and not texture information, pixel values in the background areas are very similar to one another. Thus, we can fill holes by pixel values from the background areas. This method for hole filling is quite efficient and reliable when we deal with depth images.

For the latter case, if both directional reference views are available, we can synthesize two virtual B-views using the reconstructed left and right views. In this case, the reference views are I-view and P-view or P-view and P-view. If the virtual view is synthesized by the left view, the holes occur at the left side of the foreground. Likewise, the virtual view that is synthesized from the right view has right side holes. Thus, we can fill holes by combining two virtual views. As illustrated in Fig. 4, most holes are compensated by combining two virtual views. The remaining holes are simply filled by neighboring pixels, as shown in Fig. 5.

Another problem is related to occlusion regions where more than one pixel position project into one pixel position in the target image. We refer to this problem as the overlapping problem. However, this problem can be easily solved because we deal with depth video. Among more than one pixel values of the positions, the largest depth value represents the most foreground value. Therefore, by choosing the largest values among them, we can avoid any problems.

We can obtain the final synthesized virtual view after the hole filling process. When we encode P-view and B-view, we use the virtual view as an additional reference frame using the view interpolation prediction (VIP) coding method [7]. The VIP process consists of the following five steps:

Step 1: We encode an I-view depth image, and reconstruct the I-view.

Step 2: We synthesize a virtual P-view using the reconstructed I-view, and then fill holes in the virtual P-view.

Step 3: When we encode P-view, the virtual P-view as well as the reconstructed I-view are used as reference frames. Then, we reconstruct the P-view.

Step 4: We use reconstructed I-view and P-view to synthesize two B-views which have holes in different sides. By combining two virtual B-views, we remove holes.

Step 5: When we encode the B-view, the virtual B-view is used as an additional reference frame.

IV. Experimental Results

In order to evaluate the performance of the proposed algorithm, we have implemented it into JMVM 1.0 [3]. In our experiments, we have used 100 frames of BREAKDANCERS and BALLET sequences in XGA (1024×768) format provided by Microsoft Research [8]. We arbitrarily choose the view numbers 4, 5 and 6 from eight views, where those are encoded as I-view, B-view, and P-view, respectively. When we calculate encoding bits for test sequences, we did not include any encoding bit for camera parameters.

Figure 6 to Fig. 9 and Table I through Table IV show RD curves and experimental results, respectively. Experimental results show that the overall gain of the BREAKDANCERS sequence is higher than the BALLET sequence. The reason is that the BALLET sequence has more varied depth values than BREAKDANCERS and it causes larger holes. Thus, the virtual views for BALLET are inaccurate as compared to views for BREAKDANCERS. In general, the better virtual view guarantees the better VIP coding performance. In terms of complexity, the proposed method takes about 15% more encoding time than JMVM 1.0.

In the process of synthesizing views, we have only used camera parameters. We can also fill the holes in more reliable ways than texture video due to the characteristics of the depth video. Therefore, we can say that the depth view synthesis method is performed well for multi-view depth video coding.

TABLE I. Experimental Results for "Ballet (P-view)"

	JMVM 1.0		Proposed		Gain	
QP	*PSNR (dB)*	*Bitrate (kbps)*	*PNSR (dB)*	*Bitrate (kbps)*	*Δ PSNR (dB)*	*ΔBitrate (%)*
22	47.65	549.91	47.93	504.85	+0.73	−11.42
27	44.40	341.54	44.76	319.39		
32	40.77	191.03	41.14	182.28		
37	37.43	100.72	37.90	96.30		

TABLE II. Experimental Results for "Ballet (B-view)"

	JMVM 1.0		Proposed		Gain	
QP	*PSNR (dB)*	*Bitrate (kbps)*	*PSNR (dB)*	*Bitrate (kbps)*	*Δ PSNR (dB)*	*ΔBitrate (%)*
22	47.14	416.93	47.48	399.73	+0.45	−7.61
27	43.76	256.93	44.18	252.61		
32	40.13	136.96	40.60	138.89		
37	37.13	69.71	37.55	72.02		

TABLE III. Experimental Results for "Breakdancers (P-view)"

	JMVM 1.0		Proposed		Gain	
QP	*PSNR (dB)*	*Bitrate (kbps)*	*PSNR (dB)*	*Bitrate (kbps)*	*Δ PSNR (dB)*	*ΔBitrate (%)*
22	46.98	606.03	47.26	540.88	+0.87	−16.37
27	43.58	329.01	43.90	295.37		
32	40.43	167.42	40.79	150.41		
37	37.53	85.90	37.99	78.56		

TABLE IV. EXPERIMENTAL RESULTS FOR "BREAKDANCERS (B-VIEW)"

	JMVM 1.0		Proposed		Gain	
QP	*PSNR (dB)*	*Bitrate (kbps)*	*PSNR (dB)*	*Bitrate (kbps)*	*Δ PSNR (dB)*	*ΔBitrate (%)*
22	46.37	459.83	46.73	418.64	+0.73	−14.71
27	43.06	240.97	43.45	221.62		
32	40.11	119.86	40.49	111.65		
37	37.39	62.12	37.76	59.51		

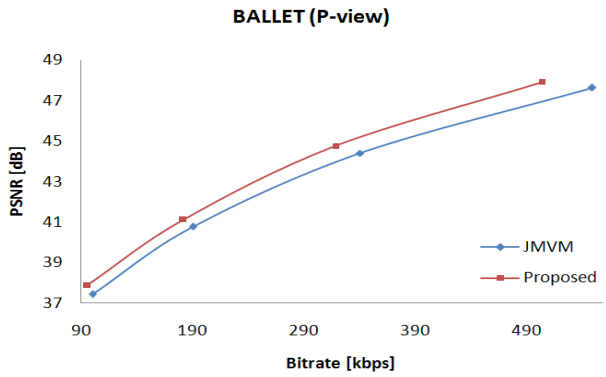

Figure 6. Rate distortion curve for BALLET (P-view)

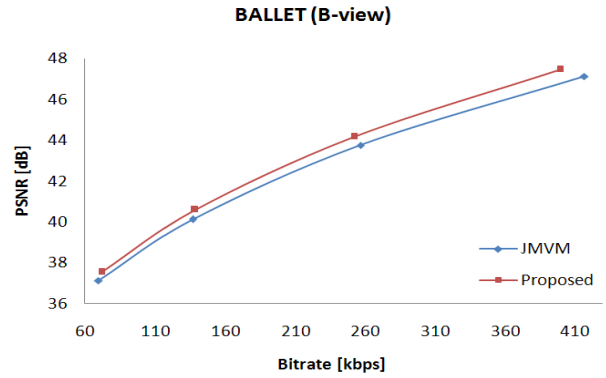

Figure 7. Rate distortion curve for BALLET (B-view)

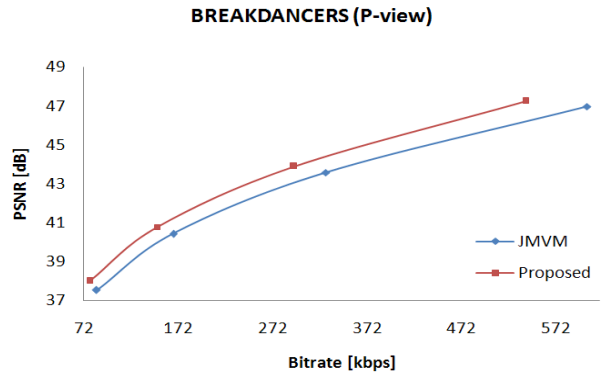

Figure 8. Rate distortion curve for BREAKDANCERS (P-view)

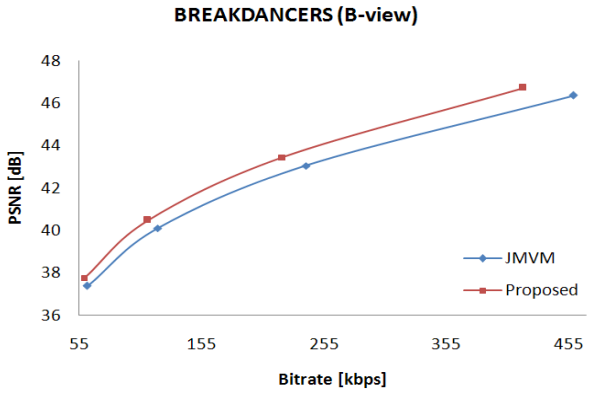

Figure 9. Rate distortion curve for BREAKDANCERS (B-view)

V. CONCLUSIONS

In this paper, we have proposed an efficient multi-view depth video coding algorithm using depth view synthesis. We have exploited the 3D warping method to synthesize a virtual view which corresponds to the current depth view. During the 3D warping process, holes are generated due to disocclusion regions. In order to solve the problem, we have used available neighboring pixels or another virtual view. Furthermore, the overlapping problem caused by occlusion regions can be avoided by choosing the largest pixel value among the overlapped pixel values. After virtual depth view synthesis, we have applied a view interpolation prediction scheme that uses the virtual current depth view as an additional reference frame in encoding the current view. We showed that the proposed algorithm achieved about 0.69 dB of PSNR gain or 12.52% bit rate reduction on average compared to JMVM 1.0

REFERENCES

[1] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D Video and Free Viewpoint Video – Technologies, Applications and MPEG Standards," Proc. Of ICME 2006, pp. 2161-2164, July 2006.

[2] ISO/IEC JTC1/SC29/WG11 N6720: Call for Evidence on Multi-view Video Coding. Oct. 2004.

[3] ISO/IEC MPEG & ITU-T VCEG JVT-U207: Joint Multiview Video Model (JMVM). Oct. 2006.

[4] ISO/IEC MPEG & ITU-T VCEG JVT-W100: Multi-view Video plus Depth (MVD) Format for Advanced 3D Video Systems. April 2007.

[5] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient Compression of Multi-view Depth Data Based on MVC," Proc. of IEEE 3DTV Conference, May 2007.

[6] S. Ince, E. Martinian, S. Yea, and A. Vetor, "Depth Estimation for View Synthesis in Multiview Video Coding," Proc. of IEEE 3DTV Conference, May 2007.

[7] C. Lee, K.J. Oh, S. H. Kim, and Y.S. Ho, "An Efficient View Interpolation Scheme and Coding Method for Multi-view Video Coding," Proc. of IWSSIP 2007, pp. 107-110, June 2007.

[8] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-qulity Video View Interpolation Using A Layered Representation," Proc. of ACM SIGGRAPH. , pp. 600-608, Aug. 2004.