Joint Coding of Multiview Video and Depth Data Using Virtual View Synthesis

Sang-Tae Na, Yo-Sung Ho

Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea

Received 30 July 2008; accepted 28 September 2010

ABSTRACT: To compress multiview video and depth information, we synthesize a virtual image for the current view using color and depth data of neighboring views. In this article, we then use a view interpolation prediction scheme at the virtual image to improve the inter-view prediction. We also propose a solution for overlapping regions and empty holes that are generated during the intermediate view synthesis process due to occlusion and disocclusion situations. Experimental results show that the proposed methods achieve approximately 0.65 dB of Peak Signal-to-Noise Ratio (PSNR) gain on average for multiview depth data and 0.17 dB of PSNR gain for multiview video coding, compared with the reference software, Joint Multiview Video Model 1.0. We also show that our method is even more powerful for smaller search ranges. Furthermore, we examine the effects of inter-view prediction on hierarchical B-pictures. Int J Imaging Syst Technol, 20, 370–377, 2010; Published online in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/ima.20260

Key words: multiview video and depth data coding; view interpolation prediction; hole filling

I. INTRODUCTION

The conventional video system can provide only one predetermined viewpoint to users. However, as the multiview video system captures the same scene with multiple cameras from different positions, it can offer arbitrary viewpoints of dynamic scenes. Hence, the system can be used in stereo TV, multiview TV as well as conventional TV. In addition, the multiview video system can be applied to advanced types of visual media, such as 3DTV and free viewpoint TV (FTV; Smolic et al., 2006). In the 3DTV, we can perceive the depth of a scene through special display devices, and FTV allows users to choose a certain viewpoint which is captured by multiple cameras or generated by using proper depth data. To support such advanced applications, we need multiview video and its corresponding depth information. However, as more cameras are used to obtain multiview scenes, the amount of multiview video and depth data also increases enormously. Therefore, an efficient coding method is required to store and transmit such huge amount of data.

In a simple way, we can compress multiview video data independently using the latest video codec, H.264/AVC (Wiegand et al., 2003). However, multiview video contains a large amount of inter-view statistical dependencies. As a result, the comparison between H.264/AVC coding and new multiview video coding (MVC) which allows inter-view prediction has been conducted in October 2004. According to the experiment, MVC shows better results than simulcast coding with H.264/AVC (ISO/IEC JTC1/SC29/WG11, 2004). To achieve better coding efficiency, in April 2006, core experiments on view-temporal prediction structure, view interpolation prediction (VIP), illumination compensation, and so on have been conducted (ISO/IEC JTC1/SC29/WG11, 2006). Consequently, Fraunhofer-Heinrich Hertz Institute (HHI) donated a software package, Joint Multiview Video Model (JMVM), in October 2006 (ISO/IEC MPEG & ITU-T VCEG, 2006). In April 2007, multiview video plus depth (MVD) format has been proposed for the advanced future video systems, including 3DTV and FTV (ISO/IEC MPEG & ITU-T VCEG, 2007).

The compression of each multiview video data and depth data has been studied for a long time. As the information between video data and its corresponding depth data is deeply correlated, trials of a joint coding of both multiview video and its corresponding depth data is recently getting attention. However, Moving Picture Experts Group (MPEG) has not yet suggested any standard for the cases. In this article, we propose a joint coding of multiview video data and its corresponding depth data. To compress multiview depth data, we synthesize a virtual view for a current view by using only the depth data themselves. We then apply the VIP scheme (Lee et al., 2007) to the virtual view to improve the inter-view prediction. Finally, we use the same procedure to MVC with the multiview depth data coded previously.

In Section II, we introduce a conventional multiview depth data coding work based on MVC. In Section III, the proposed method for multiview depth data is presented. In Section IV, we compress multiview video data using the depth data coded in Section III. In Section V, we provide experimental results to demonstrate the efficiency of our methods against the reference coding algorithm.

II. CONVENTIONAL MULTIVIEW DEPTH DATA CODING BASED ON MVC

Depth information can be acquired by two approaches. One of them is active sensor-based methods through hardware directly, such as

Correspondence to: Sang-Tae Na; e-mail: nst0721@hanmail.net
Grant sponsor: ITRC (through RBRC at GIST)

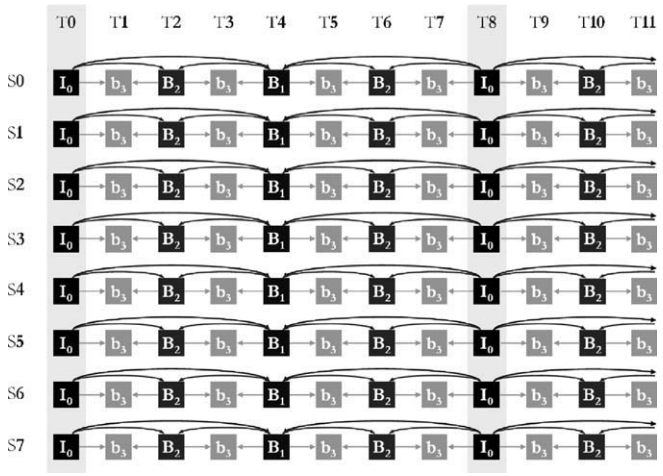

Figure 1. Simulcast coding structure with hierarchical B-pictures.

depth camera, and the other method is passive sensor-based methods using software processing, such as stereomatching (Scharstein and Szeliski, 2002).

The acquired depth information can be represented by 8 bits, where the closest point is associated with the value 255 and the most distant point with the value 0. These gray scale information can be treated as video format data. Therefore, we can use any video coding standards, such as MPEG-2, MPEG-4 visual, and H.264/AVC, to compress depth data. Moreover, the results of Fehn et al. (2002) show that the depth data can be very efficiently compressed using video coding standards. Among those video coding standards, H.264/AVC shows the highest bit saving with same video quality (Wiegand et al., 2003). Thus, H.264/AVC can be applied to each view of the multiview depth data independently, or a simulcast coding structure.

Figure 1 shows the simulcast coding structure, where S_n and T_n denote the individual view sequences and the consecutive time points, respectively. Eight cameras and a group of pictures (GOPs) length of eight are used. Also, hierarchical B-pictures are used for the temporal prediction.

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 multiview depth data coding. As the multiview depth data set consists of multiple view information of the same scene, there exists high correlation between views. Thus, spatial redundancy as well as temporal redundancy can be exploited to achieve better coding gain. As a result, the MVC method is applied identically into multiview depth data coding (Merkle et al., 2007). We have adopted same structure for our coding procedure. Figure 2 illustrates the MVC structure, which uses both temporal and spatial redundancy.

III. MULTIVIEW DEPTH DATA CODING VIA VIRTUAL VIEW SYNTHESIS

We propose a new compression method via virtual view synthesis for multiview depth data. A 3D warping operation is used to generate virtual view.

During the warping process, overlapping regions and holes are generated, so we also introduce methods to solve the problems. Afterward, the virtual view is used as additional reference frame in the encoding process.

A. Virtual View Synthesis through 3D Warping. By using camera parameters and neighboring views, we can apply a 3D warping operation to generate a virtual view which is the same view as current view. Further explanation of the 3D warping process will be followed. In textual video cases, we need extra depth information in addition to camera parameters and reference view for 3D warping. For depth data cases, however, we do not need extra bits for depth information because depth data themselves can be used as reference view as well as depth information (see Fig. 3), and this is the main reason why we apply a 3D warping operation to the multiview depth data compression. Figure 4 illustrates the overall encoder of the proposed method at the view direction.

The perspective projection matrix, which is needed for a 3D warping process, can be represented by Eq. (1).

$$\mathrm{PM} = A[R|t] \tag{1}$$

where A denotes the intrinsic parameter and R and t are the extrinsic parameters and denote a rotation matrix and translation vectors, respectively. Camera parameters consist of both the intrinsic and extrinsic parameters. By using Eq. (1) and depth information, we can obtain a projection equation, Eq. (2).

$$P_{\mathrm{ref}}(x, y, 1) \cdot D = \mathrm{PM} \cdot P_{\mathrm{WC}}(x, y, z, 1) = A[R|t] \cdot P_{\mathrm{WC}}(x, y, z, 1) \tag{2}$$

where D indicates the depth information, P_{ref} is a homogenous coordinate in the reference image coordinate system, and P_{WC} is a homogenous coordinate in the 3D world coordinate system. The equation can be transformed into Eq. (3).

$$P_{\mathrm{WC}}(x, y, z) = R^{-1} \cdot A^{-1} \cdot P_{\mathrm{ref}}(x, y, 1) \cdot D - R^{-1} \cdot t \tag{3}$$

where P_{WC} is the pixel position in the 3D world coordinate. We can project pixel positions from the homogenous coordinate in the reference image coordinate to the 3D world coordinate using Eq. (3). For the projection, we use the camera parameters from the reference view.

To synthesize a virtual view, we reproject the positions from the 3D world coordinate to the homogenous coordinate in the target

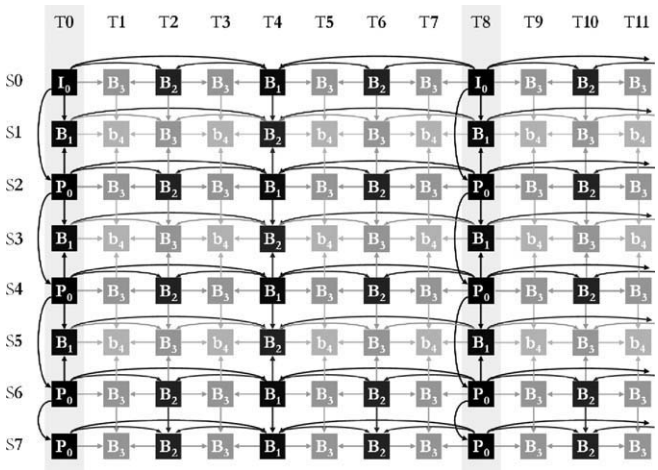

Figure 2. Multiview video coding structure with hierarchical B-pictures.

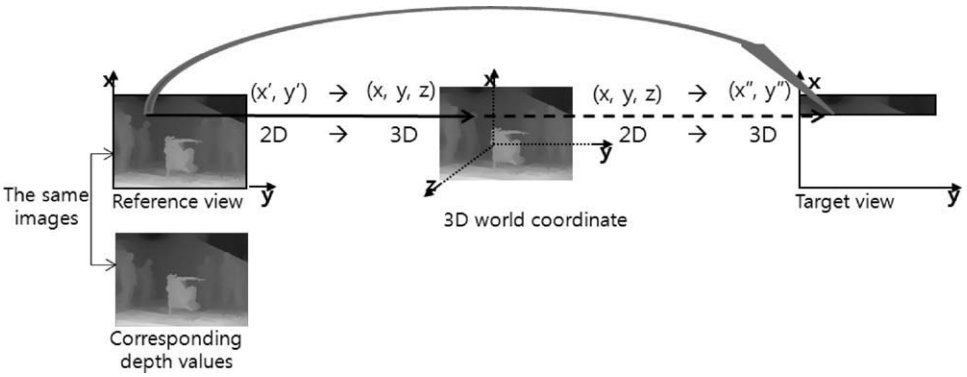

Figure 3. Virtual view synthesis process using 3D warping for depth image.

image coordinate using Eq. (4) which is the inverse equation of Eq. (3).

$$P_{\text{target}}(x, y, 1) = A \cdot R \cdot (P_{\text{WC}}(x, y, z) + R^{-1} \cdot t) \tag{4}$$

where P_{target} is a homogenous coordinate in the target image coordinate system. In this case, the camera parameters from the target view are used. Then, we get the corresponding pixel positions in the target image to the pixel positions in the reference image. Finally, we can generate a virtual view by copying the pixel values from the pixel positions in the reference image to the pixel positions in the target image.

B. Overlapping and Hole Problems. Figure 5 shows the 3D warping process at the pixel level. As explained, when we 3D warp depth images, the same depth images are used as both reference view and corresponding depth values. Thus, a pixel value located at R_1 position and a depth value of the same coordinate, D_1, are same. By using Eq. (3), pixel values at the reference view image are mapped into 3D world coordinate. Then, the pixel values are reprojected to the target view image. At the moment, some pixel values are overlapped or missed because of the occlusion and disocclusion regions. Therefore, over the 3D warping process, overlapping and hole regions are generated in the target image.

Occlusion regions are defined as areas which exist at the reference view, but invisible at the target view. In these areas, more than one pixel positions from the reference view are projected to one pixel position in the target view. Thus, pixel positions are overlapped, so pixel values are overwritten.

In the case that pixel values from background areas are overwritten by pixel values from foreground areas, we can ignore that case because it is natural that background areas are hidden by foreground areas. However, when pixel values from foreground areas are overlapped, the foreground areas are hidden by background area as shown in Figure 6(a). In this case, we can avoid the problem by choosing the biggest depth value among all the candidate values because the biggest depth value indicates the most foreground value. Figure 6(b) shows the result.

Disocclusion regions are the areas which cannot be seen at the reference view but exist at the target view. None of pixel position from the reference view is projected into the disocclusion regions of the target view, and so those areas remained as initial values and are indicated as holes. To fill the holes, we mainly use two different approaches depending on the view type.

When we have only one directional reference view, such as predictive view (P-view), we use neighboring pixel values around the hole areas. In the conventional hole filling methods, they use the interpolation scheme or preprocessed depth image (Zhang and Tam, 2005). Those methods use pixel values from both foreground and background areas to improve mainly the visual quality. However, in this article, the synthesized view is used as an additional reference frame when we encode a current view. Thus, the visual quality is not as much important as a rule of reference frame. To enhance the rule, we focus on finding the closest pixel values to the real pixel values which are supposed to be existed at the hole positions.

Most holes are generated in some regions of the background areas which are hidden by foreground areas in the reference view but are visible in the target view. Thus, we assume that holes are

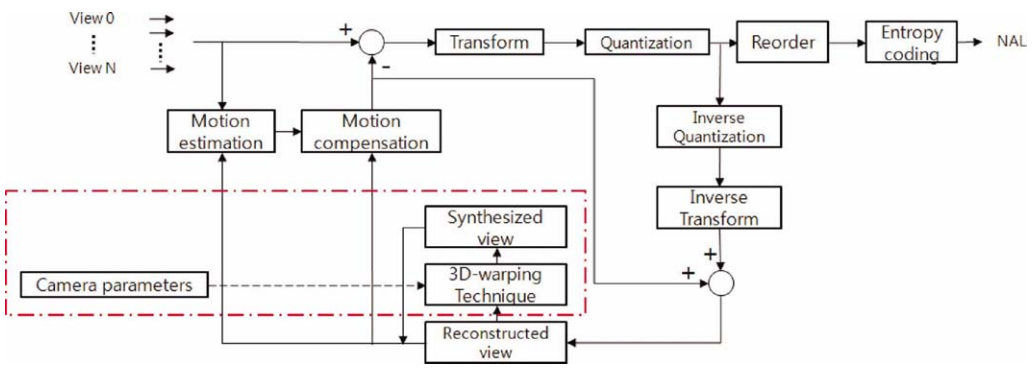

Figure 4. Multiview video coder with proposed method at the view direction. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

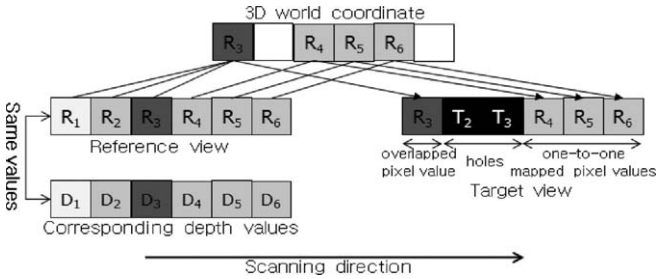

Figure 5. Pixel position projection.

mostly in the background areas. Furthermore, according to the characteristics of the depth data, pixel values are quite similar to one another because they do not have much texture and color information but only have distance information. Therefore, we fill holes using pixel values from only background areas, and again this hole filling method is quite effective when we deal with depth data.

When we have both directional reference views, such as bidirectional predictive view (B-view), we can compensate most holes by using two synthesized views which are generated by two opposite directional reference views. As shown in Figure 7, when we use a reference view from left direction, holes are generated at the left side of the foreground and the right side of the boundary of the image. And when the reference view is from right direction, holes are generated at the other side. Therefore, we can compensate most holes by combining two synthesized views and the rest holes are filled by using neighboring pixel values.

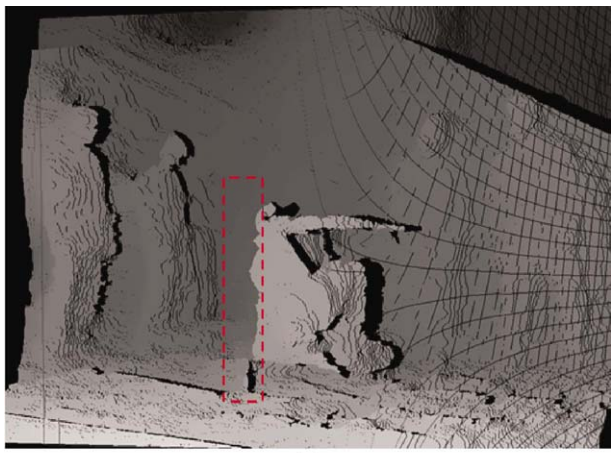

Figure 6. Overlapping problems: (a) overlapping regions, (b) without overlapping regions. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

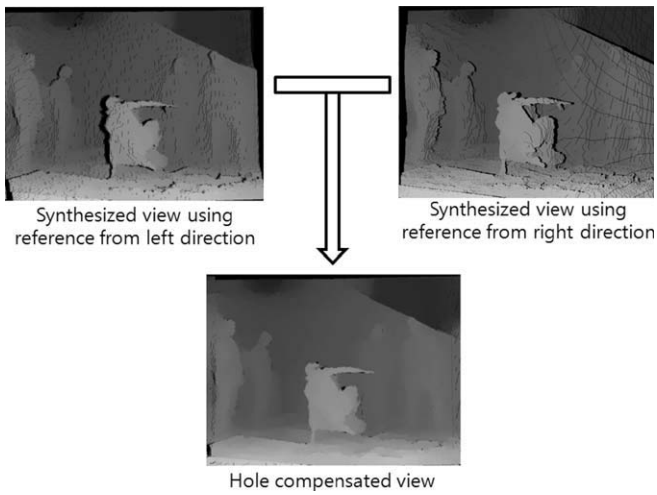

Figure 7. Hole compensation.

C. Encoding Process with a Virtual View. After completing the hole filling process, we can obtain the final synthesized view. Then, we use the synthesized view as an additional reference frame using VIP scheme (Lee et al., 2007) when we encode P- or B-view. Figure 8 illustrates the coding process.

In Figure 9, we compare the structure of the proposed algorithm to that of JMVM. The arrows indicate the direction of referring, and dotted squares are the synthesized reference frames. The numbers below each square are the coding orders.

As the quality of the 3D warping's result is highly dependent on the correctness of depth data, the better depth data correctness guarantees the better coding results in our proposal.

IV. MVC USING RECONSTRUCTED DEPTH DATA

The similar approach to multiview depth data coding is applied to MVC. As shown in Figure 10, we can generate a virtual current view using neighboring views and depth data which are already coded. For the generation, the same 3D warping operation as mentioned in the previous chapter is used. One big difference between

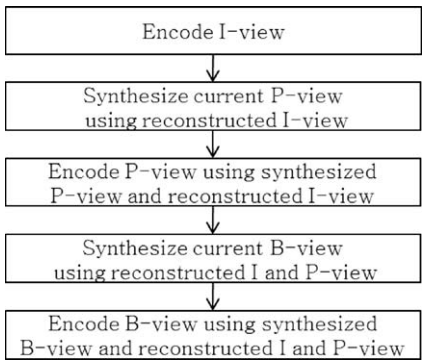

Figure 8. Coding procedure.

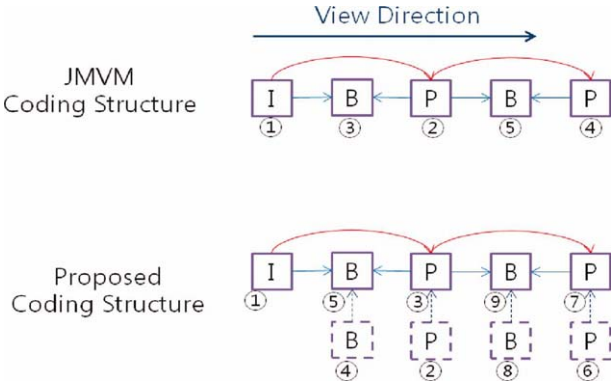

Figure 9. Coding structure comparison between JMVM and proposed algorithm. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

MVC and multiview depth data coding is that whether we need extra bits for depth information to warp or not. The latter case, we can use the same depth data themselves as reference view and depth information for 3D warping. To 3D warp for multiview video data, however, we need additional bits for depth information. Therefore, the multiview depth data coded from the previous chapter are used in the 3D warping for MVC.

A. Hole Filling using Corresponding Depth Data. Using the reconstructed depth data and neighboring views, we generate a virtual current view through Eqs. (3) and (4). In the virtual image, overlapping areas and holes are generated due to the occlusion and disocclusion regions.

To avoid overlapping problem, we use the corresponding depth data to estimate whether pixel values come from foreground areas or background areas. By choosing the pixel values which corresponded to the biggest depth value at each position as foreground pixel values, we can overcome the foreground hiding problem occurred in the occlusion regions. To fill and compensate the holes generated in disocclusion regions, we use the reconstructed depth data in addition to the hole filling methods introduced in Section IIIB. In this article, again, the synthesized virtual views are used as additional reference frames but not for visual quality. Figure 11 illustrates the synthesized image of the first frame from view number two, the depth image of the same view and frame as the synthesized image, the enlarged image of the dotted square area, and the overlapped image of (a) and (b), clockwise. As shown, the depth data can roughly indicate where holes are generated, so we can fill the holes more precisely and the effects are even higher at the planar regions.

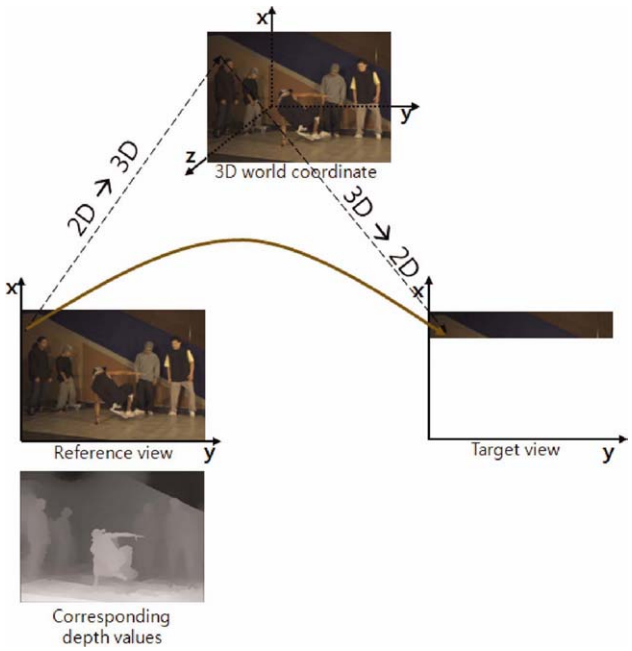

Figure 10. 3D warping for a texture image. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

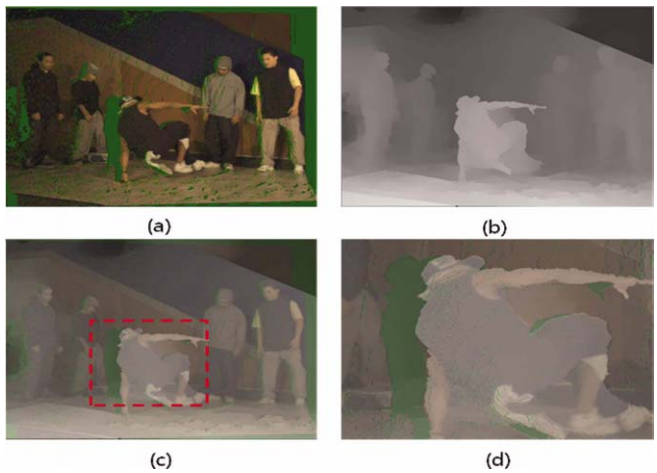

Figure 11. Hole filling with reconstructed depth data, "Breakdancers." [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

B. Search Range in the Synthesized Image. In general, different search range (SR) gives different coding gain and time. In most cases, the larger SR gives the better coding results but takes more time. Thus, we need to be careful of choosing the size of SR depending on situations and applications.

When we encode a current block in the current frame, the corresponding block in the synthesized frame is theoretically collocated. Therefore, instead of using the same SR as other reference frames, a fixed small SR can be applied to the synthesized frame to save coding time. However, it could be tricky to choose a proper SR, which depends on test sequences. From our experiments, ± 8 of SR shows the best results.

C. Possibility of Hierarchical B-pictures for Inter-view Prediction. The concept of hierarchical B-pictures applied for the temporal prediction increases the coding efficiency by providing more flexibility (Schwarz et al., 2006). Similarly, we experiment the possibility of hierarchical B-pictures for the inter-view prediction.

When the hierarchical B-picture structure is applied for the temporal prediction, the selection of the GOP length is important because it determines both coding efficiency and coding delay. In the same manner, the size of the group of views (GOVs) should be chosen carefully. When the size of the GOV is too small, the coding efficiency is not changed. On the other hand, if the size of the GOV

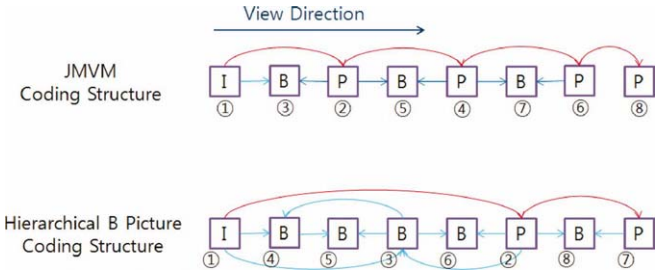

Figure 12. Coding structure comparison between JMVM and hierarchical B-pictures. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

is too big, the coding efficiency would even be degraded. Hence, the size of the GOV should be determined by considering the SR and the distance between I-view and P-view in one GOV. We then can effectively compress the last view of one GOV as P-view by referring the first view or I-view.

To satisfy the above requirement, we calculate the distance between the first view and the other views by using Eqs. (3) and (4), with the biggest depth value which moves the most by changing views. In most cases, the biggest depth value is the biggest pixel value in the gray level, 255. Among the other views, we choose one view, which makes the biggest distance within the SR, as the last view in one GOV by using Eq. (5).

$$P_{\text{ref}} - P_{\text{target}} \leq \text{ search range} \tag{5}$$

For instance, when we compress eight views of the test sequence "Breakdancers" provided by Microsoft Research (Zitnick et al., 2004), we first find the biggest depth value. In this example, as we are still on the first step of the experiment, which verifies the possibility of the hierarchical B-picture structure for inter-view prediction, we have used 16 frames of each view. Among the 16 frames of I-view, we find the biggest depth value, 203. Then, we calculate the distance between I-view and the other views using Eqs. (3) and (4), with the biggest depth value. As we select ±96 as a SR, the chosen distance should be smaller than the SR. As a result, the distance between 0-view and 5-view is 88 and 0-view and 6-view is 99, so we choose 0-view to 5-view for one GOV.

As for the last comments, since we can usually select 255 as the biggest depth value in most cases, the size of GOV depends only on the SR. Figure 12 illustrates the coding structure comparison between JMVM and hierarchical B-pictures for the inter-view prediction.

V. EXPERIMENTAL RESULTS

To verify the performance of the proposed algorithms, we implement them based on JMVM 1.0 and compare the results to the

Table I. Performance evaluation for depth data of "Breakdancers", SR ±96, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	37.36	87.93	37.77	82.83	0.73	13.66
32	40.27	166.15	40.61	153.31		
27	43.34	320.02	43.67	295.72		
22	46.68	586.60	46.99	539.71		

Table II. Performance evaluation for depth data of "Ballet", SR ±96, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	37.39	92.55	37.80	90.76	0.57	9.47
32	40.51	172.38	40.90	167.92		
27	43.98	312.91	44.33	299.63		
22	47.28	521.60	47.55	494.02		

results by the JMVM 1.0. We use "Breakdancers" and "Ballet" provided by Interactive Visual Media Group at Microsoft Research (Zitnick et al., 2004) as test sequences. Fifteen of the GOP length consisting of one I-picture, one P-picture, and 13 B-pictures is used in the sequences. Total 46 frames (three GOPs) of the sequences for all eight views are coded with various quantization parameter (QP) levels and SRs. We have experimented with 100 frames to three views and the results of Na et al. (2008) are similar to the results with 46 frames, so we use 46 frames to experiment in more various circumstances. If there is no comment, we have used exact same GOP construction as JMVM 1.0's. To calculate the average PSNR differences between rate-distortion (RD) curves and the amount of bit savings, the Bjontegaard measure is used (Bjontegaard, 2001).

A. Experimental Results by Proposed Multiview Depth Data Coding. Tables I and II show the experimental results by the proposed methods, and Figures 13 and 14 show the RD curves. As shown, the overall gain of the "Breakdancers" sequence is higher than that of the "Ballet" sequence. The reason is that the "Breakdancers" sequence has fewer scenes which contain varied depth values so they cause smaller holes and then lead to more accurate virtual views. In general, the better virtual view guarantees the better VIP coding performance. From the results, the proposed method achieved 0. 73 dB of PSNR gain from "Breakdancers" and 0.57 dB of PSNR gain from "Ballet."

B. Experimental Results for MVC. Tables III and IV show the coding results of the multiview video sequences by applying the proposed method with multiview depth data coded previously. For each QP level, we used the reconstructed multiview depth data

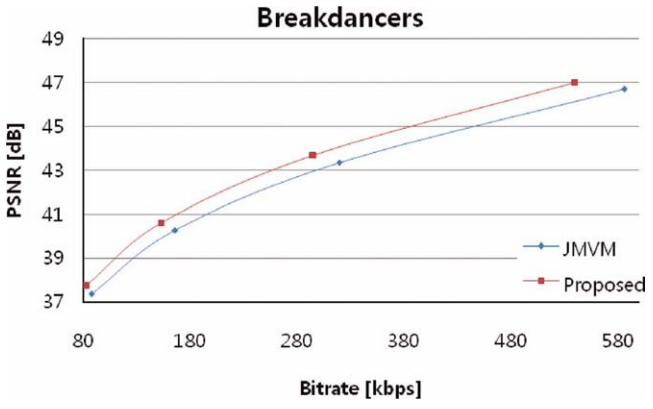

Figure 13. RD curves for depth data of "Breakdancers," SR = ±96, 46 frames. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

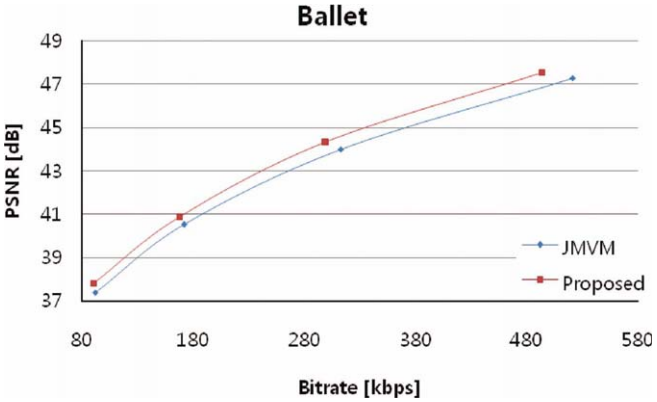

Figure 14. RD curves for depth data of "Ballet," SR = ±96, 46 frames. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table III. Performance evaluation for "Breakdancers", SR = ±96, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	35.08	127.63	35.07	116.39	0.12	4.75
32	37.17	218.84	37.12	205.12		
27	38.78	419.74	38.75	401.01		
22	40.01	979.29	40.00	942.00		

Table IV. Performance evaluation for "Ballet", SR = ±96, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	36.64	71.61	36.73	65.58	0.21	6.26
32	39.00	116.64	38.99	108.88		
27	40.91	203.36	40.89	192.86		
22	42.22	403.91	42.20	386.34		

Table V. Performance evaluation for "Breakdancers", SR = ±8, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	34.94	143.17	34.90	122.19	0.22	7.79
32	37.11	243.30	37.01	216.63		
27	38.75	456.57	38.70	423.34		
22	40.00	1038.22	39.98	991.61		

Table VI. Performance evaluation for "Ballet", SR = ±8, 46 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	36.39	79.14	36.42	69.66	0.28	7.58
32	38.88	129.74	38.85	118.56		
27	40.85	223.98	40.82	210.42		
22	42.19	433.81	42.18	416.94		

Table VII. Encoding time for "Breakdancers", SR = ±96, 16 frames.

	JMVM 1.0	Proposed	
QP	Time (s)	Time (s)	Difference (s)
37	3818	4476	658
32	4504	5378	874
27	5240	6369	1129
22	6929	8587	1658

Table VIII. Performance evaluation for "Breakdancers", SR of synthesized frames = ±8, SR of other reference frames = ±96, 16 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	35.36	122.99	35.35	122.77	0.09	0.01
32	37.31	216.77	37.31	216.62		
27	38.90	435.31	38.89	434.80		
22	40.14	1057.05	40.14	1056.78		

Table IX. Encoding time for "Breakdancers", SR of synthesized frames = ±8, SR of other reference frames = ±96, 16 frames.

	JMVM 1.0	Proposed	
QP	Time (s)	Time (s)	Difference (s)
37	4476	4301	175
32	5378	5110	268
27	6369	5972	397
22	8587	8017	570

Table X. Performance evaluation with proposed algorithm for "Breakdancers", SR = ±96, 16 frames.

	JMVM 1.0		Proposed		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	35.36	136.85	35.36	123.00	0.12	4.68
32	37.37	232.99	37.31	216.95		
27	38.92	455.78	38.90	435.56		
22	40.15	1097.86	40.14	1057.05		

Table XI. Performance evaluation with hierarchical algorithm for "Breakdancers", SR = ±96, 16 frames.

	JMVM 1.0		Hierarchical		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	35.36	136.85	34.89	121.30	0.06	2.04
32	37.37	232.99	36.99	204.57		
27	38.92	455.78	38.62	381.50		
22	40.15	1097.86	39.87	848.96		

Table XII. Performance evaluation with proposed algorithm with hierarchical for "Breakdancers", SR = ±96, 16 frames.

	JMVM 1.0		Proposed with hierarchical		Gain	
QP	PSNR (dB)	Bitrate (kbps)	PNSR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (%)
37	35.36	136.85	34.95	111.53	0.16	5.91
32	37.37	232.99	36.97	192.31		
27	38.92	455.78	38.59	366.84		
22	40.15	1097.86	39.85	827.55		

coded by the same QP level. The experimental results show 0.17 dB of PSNR gain on average. The coding gain of the multiview video data is smaller than that of the multiview depth data. Different from the multiview depth data, the multiview video data suffer from illumination difference through views. Therefore, the virtual views are not as effective as those of multiview depth data. Furthermore, the texture information also determine the quality of the virtual views so the "Ballet" sequence which has less complexity of texture information than the "Breakdancers" sequence gives better coding gain. Therefore, the coding efficiency of the proposed algorithm is dependent on the texture and color information as well as the size of holes in MVC.

C. Coding Performance with Different Search Range. In Tables III and IV, ± 96 is used as a SR, and in Tables V and VI, ± 8 is used. According to the results, the coding performance of the proposed method with the smaller SR gives better coding gain because the synthesized view is collocated with the current view. Thus, the probability of choosing the synthesized view as final reference view goes high, which leads to better coding efficiency.

D. Search Range Consideration. Table VII shows the encoding time with various QP levels when the SR is ± 96. From the table, the proposed algorithm takes about 20% more time than JMVM 1.0.

As mentioned previously, the synthesized view is the same viewpoint and frame as the current view, in other words, pixel values in both views are collocated. Therefore, we can fix the SR for the synthesized frames in a small SR. Tables VIII and IX show the comparison between when the same SR, ± 96, as the SR of other reference frames is used for synthesized frames and when the fixed small SR, ± 8, is used for synthesized frames. For this experiment, we used 16 frames of the "Breakdancers" sequence to all views. From the results, the fixed small SR gives little coding gain and contributes to reducing the coding time.

E. Hierarchical B-pictures for Interview Prediction. For the last experiment, we examine the possibility of hierarchical B-pictures for inter-view prediction. As explained in Section IV.C, 16 frames of the "Breakdancers" sequence to all views and six of the GOV length are used. Tables X–XII show the experimental results by using proposed algorithm, hierarchical B-pictures, and proposed algorithm with hierarchical B-pictures, compared with JMVM 1.0. According to the results, we find that hierarchical B-pictures implemented for inter-view prediction can also improve the coding efficiency, a little. Of course, the better hole filling methods will improve the coding efficiency more.

VI. CONCLUSIONS

We proposed an efficient compression method for combined multiview video and its corresponding depth data using virtual view synthesis. The 3D warping operation was used to generate a virtual view matched with a current view. Then, we applied VIP scheme to the virtual view to enhance the inter-view prediction during the coding process. During the synthesizing process, overlapping regions and holes are generated in the virtual view due to the occlusion and disocclusion regions. The overlapping problem is solved by using the depth information. The holes are filled by neighboring pixel values or compensated by using two synthesized views. From the experiments, we demonstrated that the proposed scheme outperforms JMVM 1.0 by 0.17 and 0.65 dB of PSNR gain on average for the multiview video data and its corresponding depth data, respectively.

REFERENCES

G. Bjontegaard, Calculation of average PSNR differences between RD-curves, 13th VCEG Meeting, Document VCEG-M33, Austin, Texas, USA, March 2001.

C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. Ijsselstekjn, M. Pollefeys, L. Vangool, E. Ofek, and I. Sexton, An evolutionary and optimized approach on 3D-TV, Int. Broadcast Convention (IBC), Amsterdam, the Netherlands, September 2002, pp. 357–365.

ISO/IEC JTC1/SC29/WG11, Call for evidence on multi-view video coding, Doc N6720, Palma de Mallorca, Spain, October 2004.

ISO/IEC JTC1/SC29/WG11, Description of Core Experiments in MVC, Doc W8019, Montreux, Switzerland, April 2006.

ISO/IEC MPEG & ITU-T VCEG, Joint Multiview Video Model (JMVM), Doc JVT-U207, Hangzhou, China, October 2006.

ISO/IEC MPEG & ITU-T VCEG, Multi-view video plus depth (MVD) format for advanced 3D video systems, Doc JVT-W100, San Jose, CA, USA, April 2007.

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, Proceedings of International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, June 2007, pp. 102–105.

P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, Efficient compression of multi-view depth data based on MVC, Proceedings of IEEE 3DTV Conference, Kos Island, Greece, May 2007.

S.T. Na, K.J. Oh, C. Lee, and Y.S. Ho, Multi-view depth video coding using depth view synthesis, IEEE International Symposium on Circuits and Systems (ISCAS), Seattle, Washington, USA, May 2008, pp. 1400–1403.

D. Scharstein and R. Szeliski, A taxonomy and evaluation of dense two-frame stereo corresponding algorithms, Int J Comput Vis 47 (2002), 7–42.

H. Schwarz, D. Marpe, and T. Wiegand, Analysis of hierarchical B pictures and MCTF, IEEE International Conference on Multimedia and Expo (ICME), Toronto, Ontario, Canada, July 2006, pp. 1929–1932.

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, IEEE International Conference on Multimedia and Expo (ICME), Toronto, Ontario, Canada, July 2006, pp. 2161–2164.

T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans Circuits Syst Video Technol 13 (2003), 560–576.

L. Zhang and W.J. Tam, Stereoscopic image generation based on depth images for 3DTV, IEEE Trans Broadcast 51 (2005), 191–199.

C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, High-quality video view interpolation using a layered representation, Proc. of ACM SIGGRAPH, Los Angeles, CA, USA, August 2004, pp. 600–608.