Depth Coding Using a Boundary Reconstruction Filter for 3-D Video Systems

Kwan-Jung Oh, Anthony Vetro, *Fellow, IEEE,* and Yo-Sung Ho, *Senior Member, IEEE*

***Abstract*—A depth image is 3-D information used for virtual view synthesis in 3-D video system. In depth coding, the object boundaries are hard to compress and severely affect the rendering quality since they are sensitive to coding errors. In this paper, we propose a depth boundary reconstruction filter and utilize it as an in-loop filter to code the depth video. The proposed depth boundary reconstruction filter is designed considering occurrence frequency, similarity, and closeness of pixels. Experimental results demonstrate that the proposed depth boundary reconstruction filter is useful for efficient depth coding as well as high-quality 3-D rendering.**

***Index Terms*—3-D video, depth boundary reconstruction filter, depth coding, view synthesis.**

I. Introduction

With the development of multimedia technology and the increasing desire for realistic media, there have been several studies on 3-D imagery [1]. A stereoscopic image [2] consisting of left and right images is able to show a realistic scene using special stereo displays. Several types of stereoscopic displays have been developed [3] and most require viewers to wear glasses to view the 3-D scene. Even though stereoscopic images provide an impressive 3-D experience, there exist further challenges to view a scene from multiple 3-D viewpoints and support auto-stereoscopic displays [4]. Auto-stereoscopic displays provide highly realistic 3-D images and free-view navigation to viewers without the need to wear glasses; this is achieved by generating various viewpoints of the scene.

Beyond the 3-D data representation, methods to efficiently compress stereoscopic images and video have been actively studied [5]. There have been several standardized approaches for compression including the MPEG-2 multiview profile [6], MPEG-4 multiple auxiliary component [7], and most recently, an extension of the H.264/AVC standard for multiview video coding (MVC) [8].

Despite the fact that multi-view video provides both 3-D perception and free view navigation, it still has some limitations to be directly used for 3-D video and free viewpoint television (FTV) systems. The performance of multi-view video systems depends significantly on the number of original views. That is, the system must capture a very large number of views and encode a huge amount of data in order to render a realistic 3-D scene with multiple viewpoints at the decoder side. However, it is difficult to acquire so many views in practical settings.

To solve the above problem, the Moving Pictures Experts Group (MPEG) has initiated work toward a new standard for 3-D video and FTV [9], [10]. This new effort, referred to as 3DVC (3-D video coding) employs a multi-view plus depth (MVD) data format instead of multi-view video only. MVD systems can represent infinite viewpoints by using virtual view synthesis techniques based on depth-image-based rendering (DIBR). The main challenges include depth estimation and virtual view synthesis, as well as 3-D video coding [11].

While MVC has been studied broadly, depth coding has received much less attention. The depth image represents a relative distance from a camera to an object in the 3-D space, and is widely used in computer vision and computer graphics fields to represent 3-D information. Most image-based rendering methods [12] utilize depth images in combination with stereo or multi-view video to synthesize the virtual view. The main objective of depth coding is to compress the depth data itself while also ensuring that the quality of a synthesized virtual view is sufficiently high. In contrast to color or texture coding, which considers the coding efficiency of the signal itself, depth coding must consider the rendering quality it is capable of producing at different rates.

Depth coding techniques could be classified into two main groups depending on the relation with the color video coding: joint coding and independent coding. Duan *et al.* [13] and Yoon *et al.* [14] proposed joint coding methods for both color and depth video based on a layered depth image. Grewatsch *et al.* [15] and Oh *et al.* [16] proposed motion vector sharing methods between color and depth image. Na *et al.* [17] proposed a view synthesis prediction for depth image coding. However, these joint depth coding techniques focused only on the improvement of depth coding efficiency itself and do not evaluate the impact on view rendering quality.

Manuscript received March 8, 2010; revised May 18, 2010 and June 26, 2010; accepted July 24, 2010. Date of publication February 17, 2011; date of current version March 23, 2011. This work was supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center Support Program supervised by the National IT Industry Promotion Agency [NIPA-2011-(C1090-1111-0003)]. This paper was recommended by Associate Editor P. Yin.

K.-J. Oh is with the Future IT Research Center, Samsung Advanced Institute of Technology, Younggin-si, Gyunggi-do 449-712, Korea (e-mail: kwanjung.oh@samsung.com).

A. Vetro is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: avetro@merl.com).

Y.-S. Ho is with the Information and Communications Department, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea (e-mail: hoyo@gist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2116590

Independent depth coding techniques encode the depth image using the characteristics of depth data and are designed by considering rendering. Morvan *et al*. [18] proposed a depth coding method using a piecewise linear function and Merkle *et al*. [19] proposed a platelet-based depth coding method. These algorithms were designed to preserve the depth boundaries since it is important for rendering. Grewatsch *et al.* [20] and Kim *et al.* [21] proposed a mesh-based depth coding. The main problem with these independent coding schemes is that even though they improved the rendering quality by strictly cutting the depth boundaries, their coding efficiency is not high.

To solve the above problem, we proposed a depth boundary reconstruction filter as an extension of our previous work in [22], which only considered occurrence frequency in the design. In this paper, we develop a new depth boundary reconstruction filter which can recover object boundaries from its neighboring pixels. Among the neighboring candidate pixels, we choose the most reliable pixel by considering the occurrence frequency, the similarity with the current pixel, and the closeness to the current pixel. This filter can recover boundaries from the noisy and smoothed image.

It is further proposed to employ this depth reconstruction filter as an in-loop filter to code the depth. Using this method in-loop can improve the rendering quality by removing noise around the object boundary and reduce the rate required to code the depth by recovering the original object boundary after it has been subject to coding loss.

We evaluate the proposed depth coding scheme with respect to the depth bit rate, depth quality, and rendering quality. To evaluate the rendering quality, we employ a virtual view synthesis method based on the DIBR [23], [24]. The rendering quality is measured by the peak signal-to-noise ratio (PSNR) value between the original image and the synthesized image. The view synthesis tool includes a depth-based 3-D warping scheme to reduce erroneous blanks, a histogram matching technique [25] to reduce the unnatural color changes in the synthesized image, a base and assistant view blending scheme to reduce the smooth and double edges artifact in the synthesized image, and finally a depth-based in-painting technique to fill the remaining holes.

The rest of this paper is organized as follows. In Section II, we analyze the depth coding. We explain the proposed depth boundary reconstruction filter and depth coding scheme in Section III and we introduce the virtual view synthesis method in Section IV. We present and analyze the experimental results in Section V. Finally, we conclude this paper in Section VI.

II. Analysis of Depth Coding

Unlike common color images, depth images exhibit a high degree of spatial correlation except at object boundaries. Thus, we can assume that the complex regions such as object boundaries are more sensitive to coding errors and need much more coding bits compared to flat regions such as backgrounds. To verify the above assumption, we encoded a depth image using an H.264/AVC encoder and then analyze the statistics of the coding results. Fig. 1 shows macroblock partitions of the coded depth image for *Ballet* sequence. Generally, the macroblocks

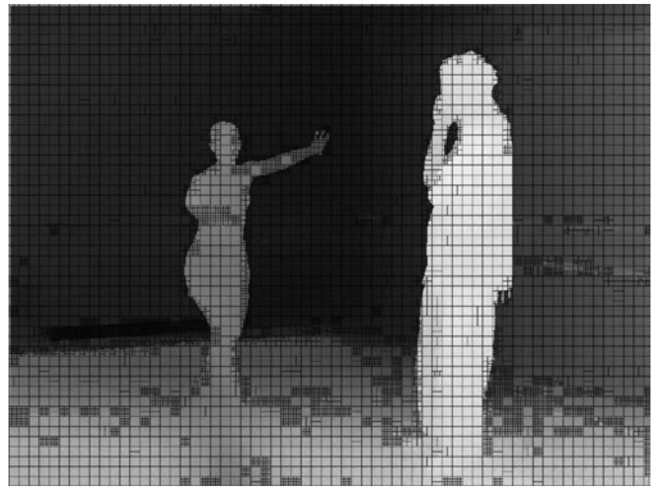


Fig. 1. Macroblock partitions for depth image of *Ballet* sequence.

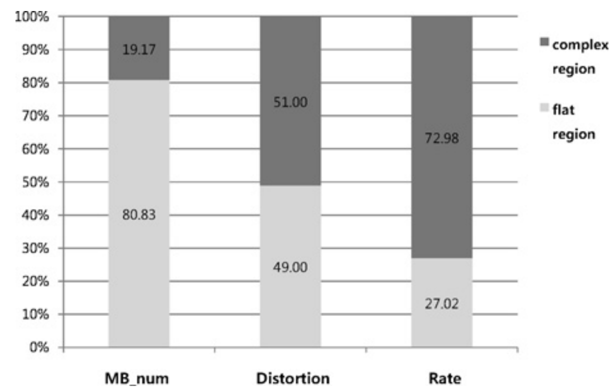


Fig. 2. Analysis on occupancy rate for number of MBs, distortion, and rate.

located on a flat region are coded by 16 × 16 macroblock partitions such as SKIP, Inter 16×16, and Intra 16×16 modes. However, the complex regions such as object boundaries are coded by more detailed partitions. On average, the flat regions need less rate and have less distortion compared to the complex regions even though flat regions occupy over 80% of the image shown in Fig. 2.

We also analyzed the sensitivity for distortion and rate by the variation of mean squared error (MSE) and average coding bits as in (1) and (2) for various quantization parameters (QP)

$$MSE_{MB} = \sum_{k=1}^{MB_{num}} \left[\frac{1}{16^2} \sum_{i=0}^{15} \sum_{j=0}^{15} \left[MB_{org}(i,j) - MB_{rec}(i,j) \right]^2 \right] \tag{1}$$

where MB_{num} is the number of macroblocks in the certain region and MB_{org} and MB_{rec} represent an original and a reconstructed macroblocks, respectively

$$Average\_bits_{MB} = \frac{1}{MB_{num}} \sum Coding\_bits \tag{2}$$

where $\Sigma Coding\_bits$ is total coding bits for a certain region.

In Fig. 3, it is shown that the MSE_{MB} value is higher in the complex region than in the flat region. In addition, the MSE value of the complex region rapidly increases in proportion to

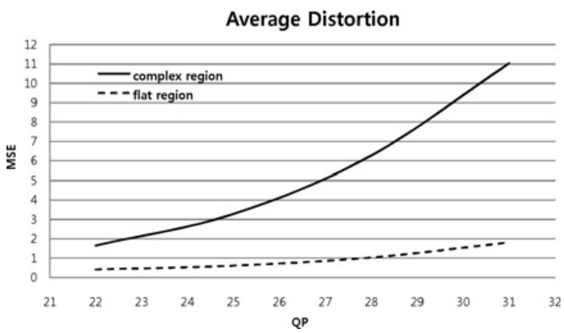


Fig. 3. Variation of the MSE in proportion to QP changes.

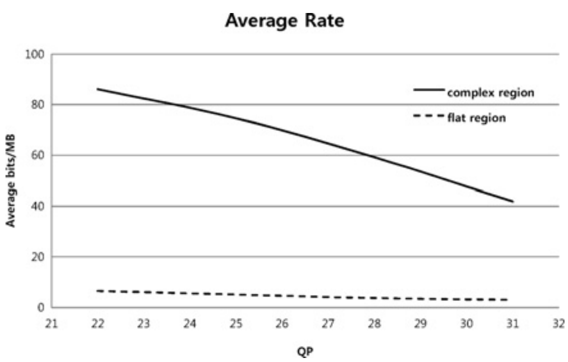


Fig. 4. Variation of the average bits per MB in proportion to QP changes.

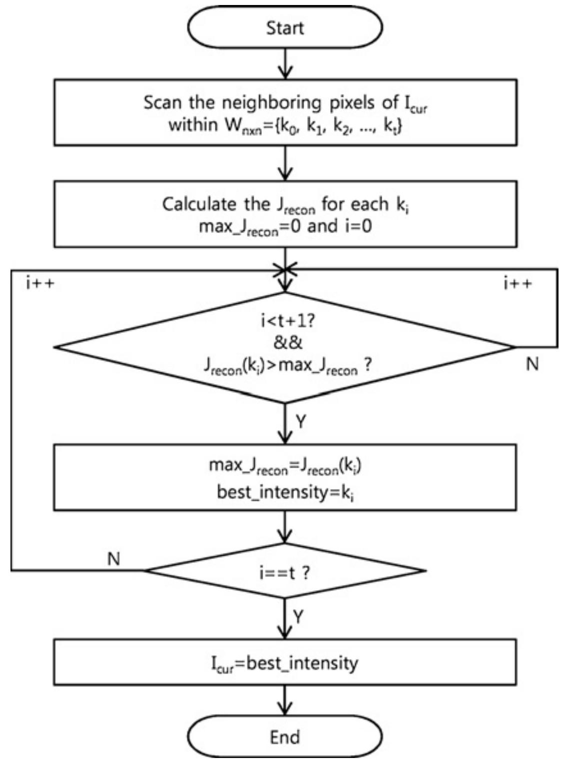


Fig. 5. Flow chart of the proposed depth boundary reconstruction filter.

QP, whereas the increase in rate is much less in the flat region.

In the same manner, we analyze the variation of the average rate; the results are shown in Fig. 4. The both rates for the complex and flat regions decrease proportional to QP, however the rate of the complex region drops rapidly. From the above observation, we conclude that even though the complex region such as the object boundary spends many coding bits, it incurs a large distortion and is sensitive to QP changes.

In general, the loss of the coded color image is regarded as acceptable if the changes of the pixel value are not recognizable by the human visual system. However, even small pixel changes in the depth image have a notable effect on the rendering quality since the depth values are used as 3-D information in the rendering process. Especially, the changes around the object boundaries severely degrade the subjective rendering quality in general.

III. Proposed Algorithms

A. Depth Boundary Reconstruction Filter

In this paper, we propose a depth boundary reconstruction filter. It is designed non-linearly by considering the following three measures: 1) occurrence frequency; 2) similarity; and 3) closeness. We define the cost function in (3) considering the above three measurements

$$J_{recon}(k) = J_F(k) + J_S(k) + J_C(k) \tag{3}$$

where k represents the pixel intensity value. We calculate the cost values for neighboring pixels of the current pixel and find the best intensity value which has the maximum cost value. That is, the current pixel is replaced with the best intensity value. The neighborhood is restricted by an $n \times n$ window (W_{nxn}), where n is an odd number. Fig. 5 illustrates a flow chart of our proposed depth boundary reconstruction filter.

The first sub-cost function, J_F, stands for the occurrence frequency for each intensity value. It is derived from the number of occurrences (N_{oc}). We calculate N_{oc} for a certain intensity value (k) in W_{nxn} using

$$N_{oc}(k) = \sum_{i=0}^{(n\times n-1)} \delta[k, W_{n\times n}(i)],\ \delta[a,b] = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $W_{nxn}(i)$ denotes the intensity value for pixel i in W_{nxn}. Then, $J_F(k)$ is calculated as given in

$$J_F(k) = \frac{N_{oc}(k) - N_{oc}(\min)}{N_{oc}(\max) - N_{oc}(\min)} \tag{5}$$

where $N_{oc}(\max)$ and $N_{oc}(\min)$ represent the maximum and the minimum N_{oc} values, respectively.

The second sub-cost function, J_S, means the similarity (S) of intensity between the current pixel (I_{cur}) and its neighborhood (I_k). It is measured by the absolute difference as given in

$$S(k) = |I_{cur} - I_k|\,. \tag{6}$$

By using (6), we calculate $J_S(k)$ as depicted in

$$J_S(k) = \frac{S(\max) - S(k)}{S(\max) - S(\min)} \tag{7}$$

where $S(\max)$ and $S(\min)$ indicate the maximum and the minimum similarity values, respectively.

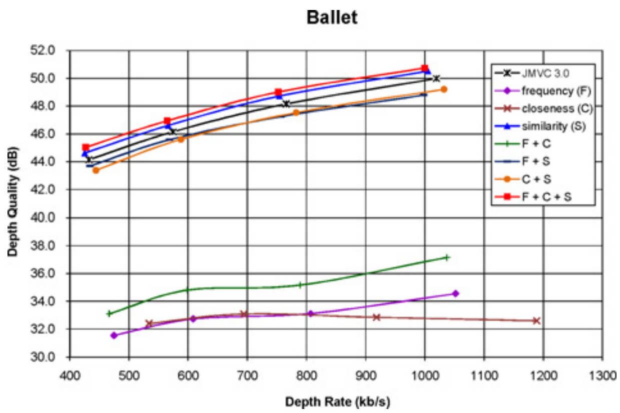


Fig. 6. Comparison of various depth boundary reconstruction filters.

The third sub-cost function, J_C, represents the closeness (C) between the current pixel (x_{cur}, y_{cur}) and its neighborhood (x_i, y_i). It is measured by the average Euclidean distance as given in

$$C(k) = \frac{1}{N_{oc}(k)} \sum_{i=0}^{N_{oc}(k)-1} \sqrt{(x_{cur} - x_i)^2 + (y_{cur} - y_i)^2} \quad (8)$$

where (x_{cur}, y_{cur}) and (x_i, y_i) are pixel coordinates of the current pixel and the pixel with intensity value k, respectively. By using $C(k)$ in (8), we calculate the $J_C(k)$ as noted in

$$J_C(k) = \frac{C(\max) - C(k)}{C(\max) - C(\min)} \quad (9)$$

where $C(max)$ and $C(min)$ are the maximum and the minimum closeness values, respectively.

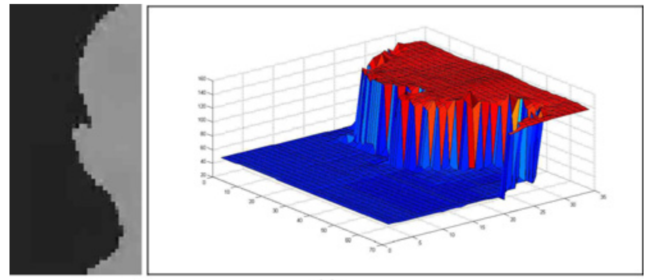
The following experimental result shows the rationale for composing the proposed depth boundary reconstruction filter with the combination of three measurement costs. We tested several types of depth boundary reconstruction filters consisting of various combinations of the three costs. As shown in Fig. 6, the depth boundary reconstruction filter that accounts for all three costs shows the best result.

The proposed depth boundary reconstruction filter has the following advantages over other linear filters: 1) it is more robust against outliers, a single pixel that does not have a similar intensity to those of neighboring pixels will not affect filtering significantly, and 2) since the filtered value must actually be the value of one of the pixels in the neighborhood, filtering does not create new unrealistic pixel values. Fig. 7 illustrates example images for the proposed depth boundary reconstruction filter.

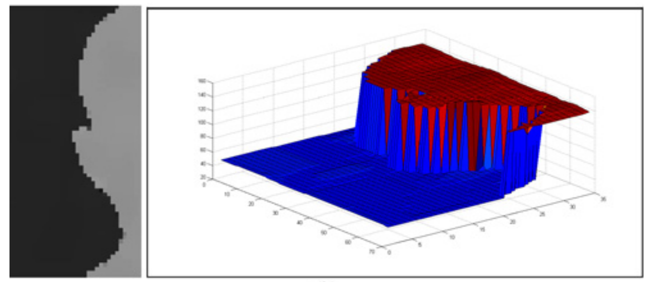
The proposed depth boundary reconstruction filter is robust to coding errors and blurring. However, some Gaussian noise still remains after filtering. To eliminate the remaining errors, we apply the 3 × 3 bilateral filter [26]. The bilateral filter is a representative non-linear edge-preserving filter. The bilateral filter extends the concept of Gaussian smoothing by weighting the filter coefficient with the intensity value of the corresponding relative pixel. Pixels that are very different in intensity from the central pixel are weighted less even though they may be in close proximity to the central pixel. This is a convolution with a non-linear Gaussian filter having weights based on pixel intensities and locations as given in

$$BF\,[I_p] = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(||p - q||) \ \ G_{\sigma_r}\left(|I_p - I_q|\right) I_q \quad (10)$$

where I_p is the central pixel and I_q is the neighboring pixel of I_p in the window W_p. BF[I_p] is the bilateral filtered value for the central pixel. The filter parameters, space sigma (σ_s) and range sigma (σ_r), define the spatial extent of the kernel and threshold for color differences in the image, respectively.



(a)



(b)

Fig. 7. Result images for the depth boundary reconstruction filter. (a) Before depth boundary reconstruction. (b) After depth boundary reconstruction.

B. Depth Coding with In-Loop Filter

The coding errors for the depth image are mainly caused by the quantization process. In typical video coding schemes, the residual data between an original image and its predicted image are transformed to the frequency domain and then quantized. The quantization process reduces the encoding bits by reducing the entropy of the transform coefficients. The amount of loss is adjusted by the QP value and the coding errors increase proportional to the QP value. In the case of depth coding, most coding errors are concentrated in high frequency region such as object boundaries as discussed in Section II.

The proposed depth boundary reconstruction filter shows good performance as shown in the previous subsection. To utilize the boundary reconstruction filter for depth coding, we design a new depth coding scheme. In this scheme, we use the depth boundary reconstruction filter as an in-loop filter similar to the existing deblocking filter in H.264/AVC. Fig. 8 shows the block diagram of the in-loop depth coding scheme in the context of an H.264/AVC codec using proposed depth boundary reconstruction filter.

In the in-loop depth coding scheme, the depth boundary reconstruction filter is located right after the deblocking filter. It improves the quality of the current frame by removing the

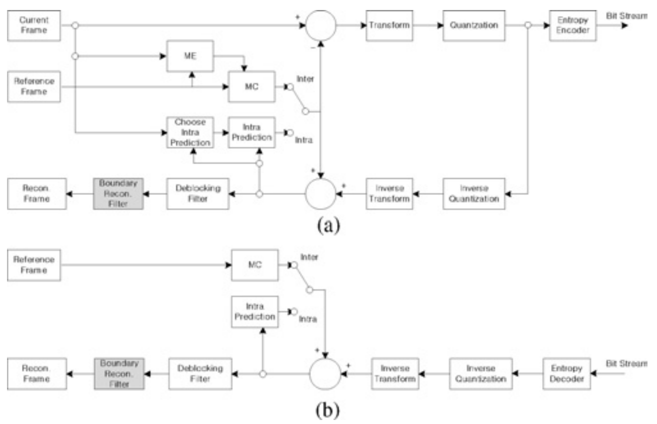


Fig. 8. Block diagram of the in-loop depth coding framework for the codec compatible with H.264/AVC. (a) Encoder. (b) Decoder.

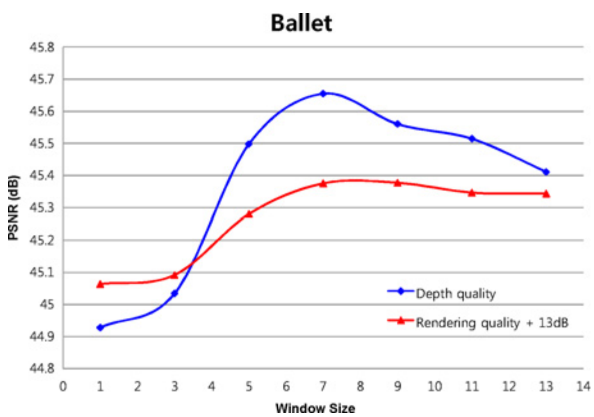


Fig. 9. Variation of depth quality and rendering quality for window size.

noise around the object boundary and reduces the encoding bits of the next frame by increasing the correlation between the next frame and the current reconstructed frame. The in-loop depth coding scheme is expected to improve both the depth coding efficiency and the rendering quality.

C. Determination of Filter Parameters

The depth boundary reconstruction filter is designed to recover the depth boundaries from smoothed images as well as noisy images. However, its efficiency highly depends on the filter window size as shown in Fig. 9. In general, better depth quality yields better rendering quality.

For the bilateral filter, the effect of filtering is mainly determined by the range sigma value rather than the space sigma value. Fig. 10 shows the influence of variation of the range sigma when the space sigma is fixed to one. In this case, a better depth quality does not ensure a better rendering quality.

With the proposed boundary reconstruction filter, we assume that the window size and range sigma having the best depth quality would guarantee the best rendering quality. We find the best window size and range sigma by considering depth quality for each slice as shown in Fig. 11. This information is then encoded into the bitstream using the codewords in Table I. The total rate overhead is six bits per slice.

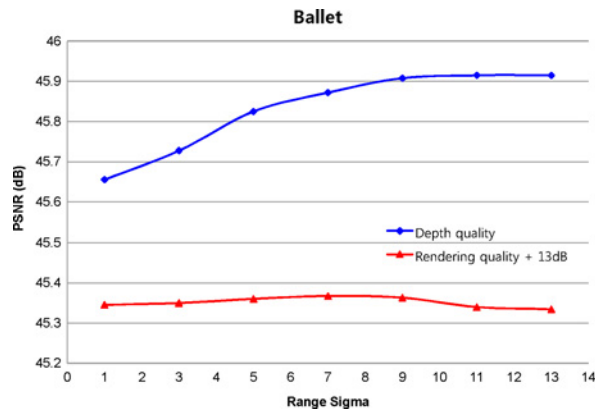


Fig. 10. Variation of depth quality and rendering quality for range sigma.

TABLE I
CODEWORD TABLE FOR WINDOW SIZE AND RANGE SIGMA

Codeword	000	001	010	011	100	101	110	111
window_size	1	3	5	7	9	11	13	15
range_sigma	1	3	5	7	9	11	13	15

```
slice_data( ){
   ⋮
   macroblock_layer( ){
   ⋮
   }
      window_size
      range_sigma
      end_of_slice_flag
   ⋮
}
```

Fig. 11. Syntaxes for window size and range sigma in slice data syntax.

IV. VIEW SYNTHESIS AND EVALUATION OF DEPTH CODING

A. Virtual View Synthesis

The virtual view synthesis is a key technology in 3-D video systems and there are many rendering methods. In this paper, we use our view synthesis method [27]. It was designed based on the view synthesis tool [28] that is being used in 3DVC group. The overall procedure of the proposed virtual view synthesis is shown in Fig. 12.

First, we conduct 3-D warping for two reference views. In 3-D warping, pixels in the reference image are back-projected to 3-D spaces, and re-projected onto the virtual view. For that, the depth image for the reference views and the camera parameters for three views are used. The back-projection and re-projection processes are represented in (11) and (12), respectively

$$(x, y, z)^T = R_{ref} A_{ref}^{-1} (u, v, 1)^T d_{u,v} + t_{ref} \tag{11}$$

$$(l, m, n)^T = A_{ver} R_{ver}^{-1} \{(x, y, z)^T - t_{ver}\} \tag{12}$$

where A, R, and t are camera parameters and d denotes the depth value. The coordinate (u, v) located on the reference

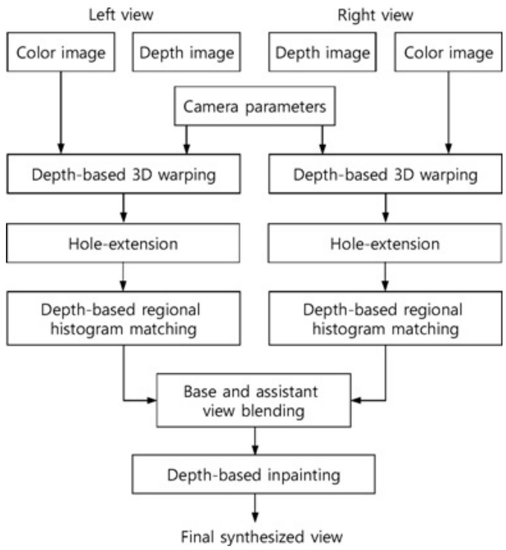


Fig. 12. Overall procedure of the virtual view synthesis.

TABLE II
EXPERIMENTAL RESULTS FOR *Breakdancers* SEQUENCE

QP	Depth Rate (kb/s)			Depth Quality (dB)		
	JMVC 3.0	Method 1	Method 2	JMVC 3.0	Method 1	Method 2
22	1174.53	1162.30	1157.39	49.85	50.43	50.73
25	863.78	856.99	852.36	47.83	48.38	48.62
28	616.88	615.39	612.16	45.69	46.16	46.36
31	432.15	434.20	431.24	43.55	43.93	44.09
BD gain	Method 1: 0.53 dB or 7.84%			Method 2: 0.78 dB or 11.28%		

TABLE III
EXPERIMENTAL RESULTS FOR *Ballet* SEQUENCE

QP	Depth Rate (kb/s)			Depth Quality (dB)		
	JMVC 3.0	Method 1	Method 2	JMVC 3.0	Method 1	Method 2
22	943.50	932.06	925.30	49.93	50.44	50.68
25	705.07	696.76	690.35	48.13	48.72	48.91
28	526.68	522.10	515.50	46.13	46.76	46.92
31	391.52	393.23	387.38	44.02	44.69	44.91
BD gain	Method 1: 0.66 dB or 9.43%			Method 2: 0.93 dB or 12.88%		

view is 3-D warped to (U, V) on the virtual view. The coordinate (l, m, n) in (12) is normalized to $(l/n, m/n, 1)$ and then represented as an integer-coordinate (U, V) in the virtual view.

To avoid the errors such as black-contours which appear in the 3-D warped images, a 3-D warping of the depth image is performed first followed by a median filtering to remove the block-contour errors. After that, we copy the corresponding color pixels from the reference view. The boundary trace-errors around the big holes are generally caused by inaccurate boundary matching between the color images and depth images. To remove these visible errors, we extend the hole's boundaries by using image dilation [29]. The extended holes can be filled by the other 3-D warped view and we can expect a more natural synthesized view by removing this kind of errors.

Before a view blending of two 3-D warped images, we apply a histogram matching to reduce the color differences between the two 3-D warped images. However, this causes unnatural effects in synthesized images. Based on the histogram matching algorithm described in [25], we modify the 3-D warped images to have the same holes to improve the accuracy of the histogram matching and virtually define the histogram for a virtual view by averaging the results of the left and right views. Then, we compensate the color changes between the virtual view and the reference view.

The next step is view blending, which combines the two 3-D warped views into the virtual view. We define the closer reference view as the base view and the other reference view as the assistant view. The base view (I_B) is manly used to synthesize the virtual view (I_V) and the assistant view (I_A) is used to fill up the holes of the base view as depicted in

$$I_V(x, y) = \begin{cases} I_B(x, y), \text{ if } I_B(x, y) \notin holes \\ I_A(x, y), \text{ otherwise} \end{cases} \tag{13}$$

where (x, y) is a coordinate in the image.

Even though the above view blending method efficiently fills up most holes, some holes still remain. These remaining holes are caused by remaining disoccluded regions or initially wrong depth values. The disocclusion region is defined as an area that cannot be seen in the reference image, but exists in the synthesized one. Most of the existing hole-filling methods use image interpolation or in-painting techniques [30] and fill up the remaining holes using neighboring pixels solely based upon geometrical distance. However, the hole filling using the background pixels rather than the foreground ones is more reasonable by the definition of the disocclusion. Therefore, we proposed to fill up the remaining holes using the depth based in-painting technique [31].

B. Evaluation of Depth Coding

The coding efficiency of the image is measured in rate and distortion metrics. In general, the encoding bit rate and PSNR value are used as the rate and distortion measures, respectively. The PSNR value is calculated by using MSE between the original image and reconstructed image. However, because the depth image is 3-D information to help the virtual view synthesis, its quality should be also evaluated in rendering quality. In this paper, we employ the $PSNR_{ren}$ as in (14) between the existing original image (I_{org}) and the rendered image (I_{ren}). The MSE_{ren} is defined as in (15)

$$PSNR_{ren} = 10 \times \log_{10}\left(\frac{255^2}{MSE_{ren}}\right) \tag{14}$$

$$MSE_{ren} = \frac{1}{w \times h}\sum_{i=0}^{w-1}\sum_{j=0}^{h-1}\left|I_{org}(i, j) - I_{ren}(i, j)\right|^2. \tag{15}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

We have tested the proposed algorithm with *Breakdancers* and *Ballet* sequences [32]. Among the 8 views, view 3 and view 5 were selected as reference views and view 4 was set as the virtual view. We encoded three views with 100 frames using JMVC 3.0 [33] with QP 22, 25, 28, and 31. The delta QPs, differential QP between the basis layer and sub-layer in

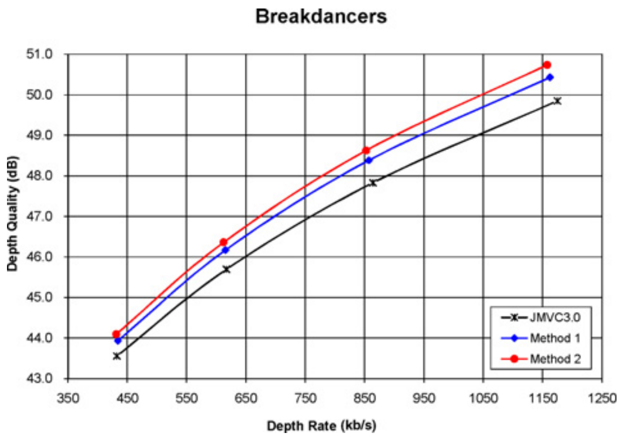


Fig. 13. RD curves in terms of depth rate and depth quality (*Breakdancers*).

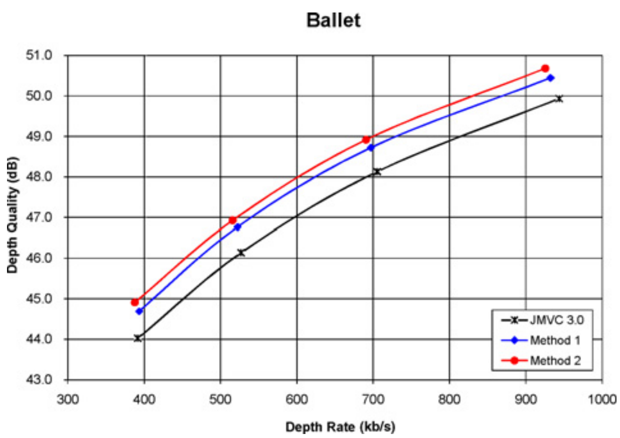


Fig. 14. RD curves in terms of depth rate and depth quality (*Ballet*).

the hierarchical-B picture structure [34], were set to zero in all layers. In addition, we also tested the *Beergarden* sequence [35] with 49 frames but did not test rendering quality for that since it only provides a depth video for a single view. We used the view synthesis method in Section IV and the color videos for rendering were used without coding.

The efficiency of the proposed method is evaluated in terms of depth coding itself and the evaluation method in Section IV-B. The JMVC 3.0 coding method in the experimental results refers to the depth coding results without the proposed method. We tested two depth boundary reconstruction filters. Method 1 denotes the proposed depth boundary reconstruction filter and Method 2 represents the combination of Method 1 and the 3×3 bilateral filter. Tables II–IV show depth coding results for three test sequences.

As shown in the above results, the proposed depth boundary reconstruction filter and its combination with bilateral filter achieved PSNR gains of 0.54 dB and 0.71 dB, respectively, in terms of average Bjontegaard metric [36] over all three sequences, which translates to a bit rate savings of 9.09% and 11.33%. The rate-distortion (RD) curves in terms of depth rate and depth quality are shown in Figs. 13–15.

Tables V and VI show experimental results for rendering quality. In terms of depth rate versus rendering quality, the proposed depth boundary reconstruction filter and its extended version which combines with bilateral filter achieved PSNR gains of 0.15 dB and 0.14 dB, respectively, in terms of average Bjontegaard metric [36] over *Breakdancers* and *Ballet* sequences, which translates to a bit rate savings of 41.79% and 36.17%. Generally, the simple model was better than combination with the bilateral filter in terms of depth bit rate and rendering quality, especially at the low bit rate. This is because that the detail regions are smoothed by the bilateral filter.

The RD curves in terms of depth rate and rendering quality are shown in the Figs. 16 and 17. As shown in Figs. 18 and 19, the subjective quality of the rendered image is also improved.

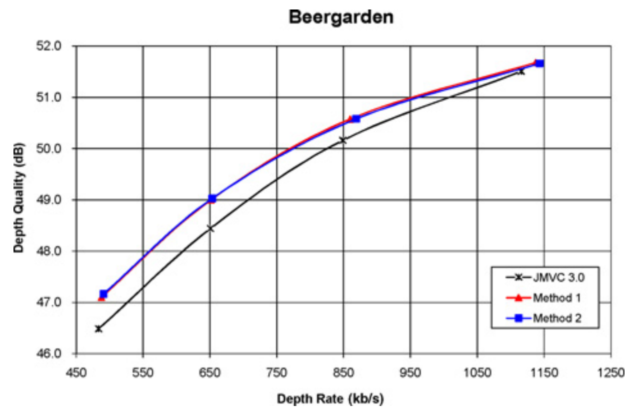


Fig. 15. RD curves in terms of depth rate and depth quality (*Beergarden*).

TABLE IV
EXPERIMENTAL RESULTS FOR *Beergarden* SEQUENCE

QP	Depth Rate (kb/s)			Depth Quality (dB)		
	JMVC 3.0	Method 1	Method 2	JMVC 3.0	Method 1	Method 2
22	1115.69	1137.16	1143.09	51.51	51.68	51.66
25	848.49	859.71	868.55	50.16	50.57	50.58
28	650.71	651.46	653.58	48.44	49.00	49.03
31	483.32	487.72	490.83	46.48	47.10	47.16
BD gain	Method 1: 0.42 dB or 7.00%			Method 2: 0.41 dB or 9.83%		

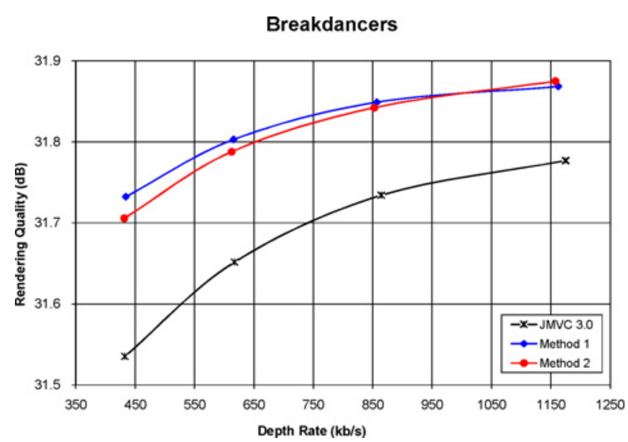


Fig. 16. RD curves in terms of depth rate and rendering quality (*Breakdancers*).

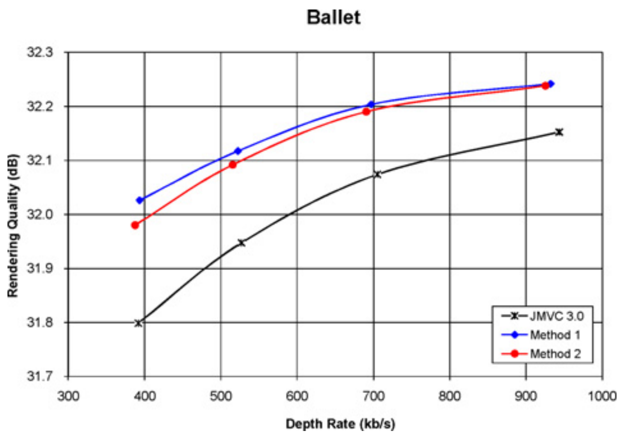


Fig. 17. RD curves in terms of depth rate and rendering quality (*Ballet*).

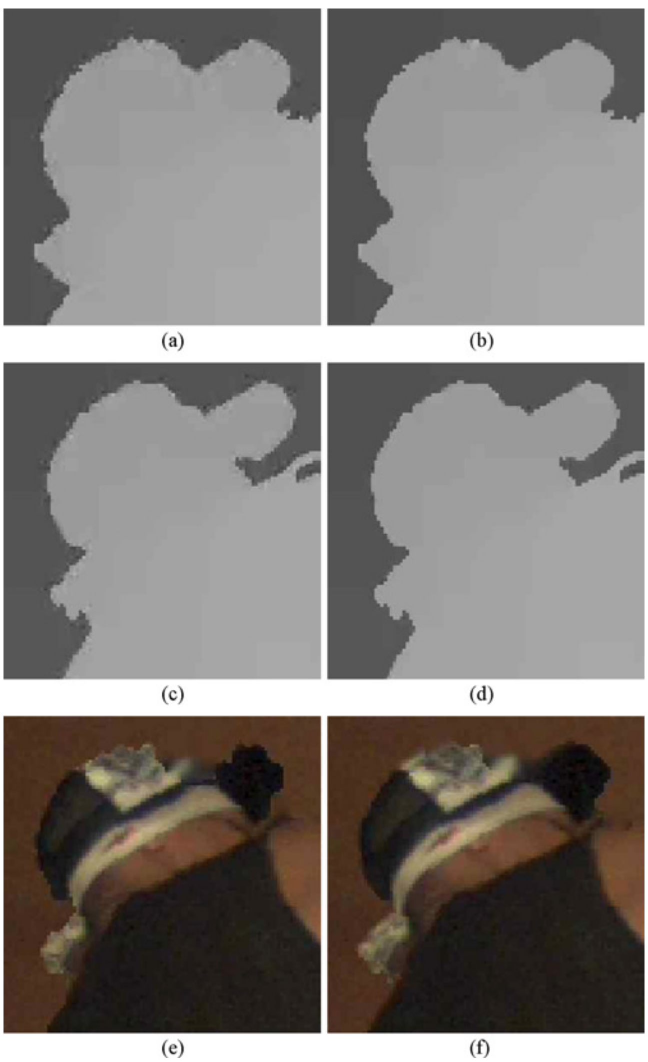


Fig. 18. Reconstructed depth images and rendered images for *Breakdancers* sequence coded with QP 31 (first frame). (a) Left reference depth image by JMVC 3.0. (b) Left reference depth image by proposed method. (c) Right reference depth image by JMVC 3.0. (d) Right reference depth image by proposed method. (e) Synthesized image from (a) and (c). (f) Synthesized image from (b) and (d).

TABLE V

EXPERIMENTAL RESULTS FOR *Breakdancers* SEQUENCE

QP	Depth Rate (kb/s)			Rendering Quality (dB)		
	JMVC 3.0	Method 1	Method 2	JMVC 3.0	Method 1	Method 2
22	1174.53	1162.30	1157.39	31.78	31.87	31.87
25	863.78	856.99	852.36	31.73	31.85	31.84
28	616.88	615.39	612.16	31.65	31.80	31.79
31	432.15	434.20	431.24	31.53	31.73	31.71
BD gain	Method 1: 0.14 dB or 47.63%			Method 2: 0.13 dB or 41.14%		

TABLE VI

EXPERIMENTAL RESULTS FOR *Ballet* SEQUENCE

QP	Depth Rate (kb/s)			Depth Quality (dB)		
	JMVC 3.0	Method 1	Method 2	JMVC 3.0	Method 1	Method 2
22	943.50	932.06	925.30	32.15	32.24	32.24
25	705.07	696.76	690.35	32.07	32.20	32.19
28	526.68	522.10	515.50	31.95	32.12	32.09
31	391.52	393.23	387.38	31.80	32.03	31.98
BD gain	Method 1: 0.16 dB or 35.94%			Method 2: 0.14 dB or 31.20%		

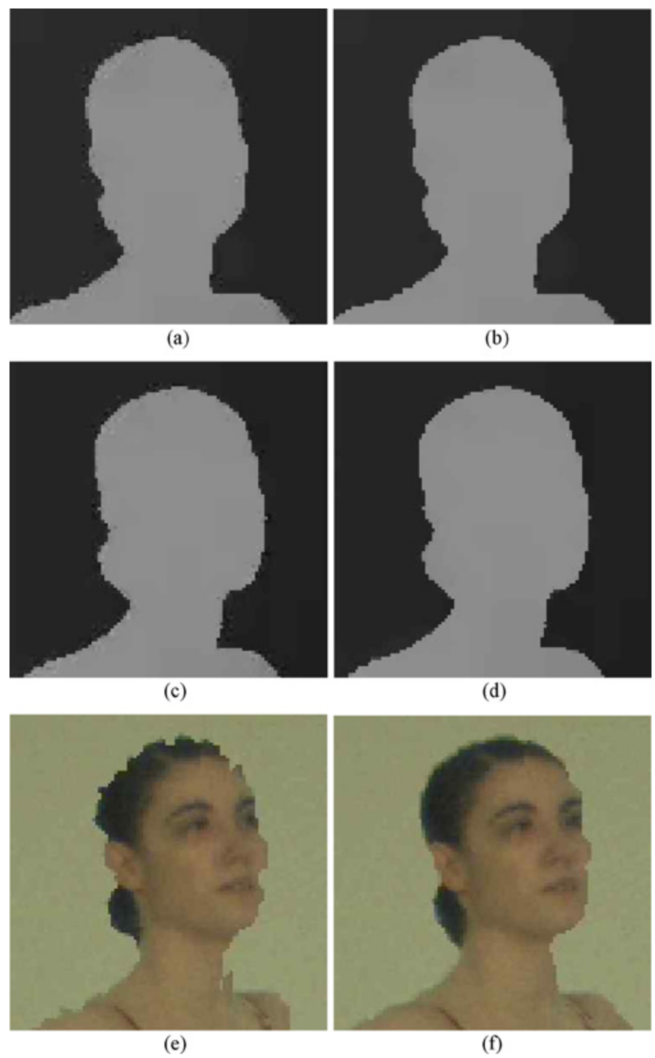


Fig. 19. Reconstructed depth images and rendered images for *Ballet* sequence coded with QP 31 (31st frame). (a) Left reference depth image by JMVC 3.0. (b) Left reference depth image by proposed method. (c) Right reference depth image by JMVC 3.0. (d) Right reference depth image by proposed method. (e) Synthesized image from (a) and (c). (f) Synthesized image from (b) and (d).

VI. Conclusion

In this paper, we proposed the depth coding method using a boundary reconstruction filter which was designed considering the occurrence frequency, the similarity, and the closeness. The proposed depth boundary reconstruction filter is robust to noise and smoothness. By using this property, we proposed the depth coding tool. We designed the coding scheme employing the depth boundary reconstruction filter as an in-loop filter. We evaluated the efficiency of the proposed depth coding scheme in terms of depth bit rate, depth quality, and rendering quality. From experimental results, we confirmed that the proposed method reduces the depth rate and improves the rendering quality including subjective quality.

Acknowledgment

The authors acknowledge the contribution of S. Yea to an earlier version of this paper.

References

[1] A. Smolic and D. McCutchen, “3DAV exploration of video-based rendering technology in MPEG,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 348–356, Mar. 2004.

[2] F. G. Waack, *Stereo Photography* (English translation of German book). German Stereoscopic Society, 1985.

[3] T. Saishu, S. Numazaki, K. Taira, R. Fukushima, A. Morishita, and Y. Hirayama, “Flatbed-type autostereoscopic display system and its image format,” *Proc. SPIE*, vol. 6055, pp. 261–268, 2006.

[4] W. Matusik and H. Pfister, “3-D TV: A scalable system for real-time acquisition, transmission and autostereoscopic display of dynamic scenes,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 814–824, Aug. 2004.

[5] S. Sun and S. Lei, “Stereo-view video coding using H.264 tools,” *Proc. SPIE Image Video Commun. Process.*, vol. 5685, pp. 177–184, Mar. 2005.

[6] X. Chen and A. Luthra, “MPEG-2 multiview profile and its application in 3-D TV,” *SPIE-Multimedia Hardw. Architectures*, vol. 3021, pp. 212–223, Feb. 1997.

[7] H. A. Karim, S. Worrall, A. H. Sadka, and A. M. Kondoz, “3-D video compression using MPEG-4-multiple auxiliary component (MPEG4-MAC),” in *Proc. 2nd IEE Int. Conf. VIE*, Apr. 2005.

[8] *Text of ISO/IEC 14496-10:2008/FDAM 1 Multiview Video Coding*, document w9978, ISO/IEC JTC1/SC29/WG11, Oct. 2008.

[9] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” *Proc. IEEE*, vol. 93, no. 1, pp. 99–110, Jan. 2005.

[10] F. Isgro, E. Trucco, P. Kauff, and O. Schreer, “Three-dimensional image processing in the future of immersive media,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 288–303, Mar. 2004.

[11] M. Tanimoto, “Overview of free viewpoint television,” *Signal Process. Image Commun.*, vol. 21, pp. 454–461, Jul. 2006.

[12] S. C. Chan, H. Y. Shum, and K. T. Ng, “Image-based rendering and synthesis,” *IEEE Signal Process. Mag.*, vol. 24, no. 6, pp. 22–33, Nov. 2007.

[13] J. Duan and J. Li, “Compression of the layered depth image,” *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 365–372, Mar. 2003.

[14] S. U. Yoon and Y. S. Ho, “Multiple color and depth video coding using a hierarchical representation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1450–1460, Nov. 2007.

[15] S. Grewatsch and E. Mûller, “Sharing of motion vectors in 3-D video coding,” in *Proc. ICIP*, vol. 5. Oct. 2004, pp. 3271–3274.

[16] H. Oh and Y. S. Ho, “H.264-based depth map sequence coding using motion information of corresponding texture video,” in *Proc. Pac. Rim Symp. Adv. Image Video Technol.*, Dec. 2006, pp. 898–907.

[17] S. T. Na, K. J. Oh, and Y. S. Ho, “Joint coding of multi-view video and corresponding depth map,” in *Proc. ICIP*, Oct. 2008, pp. 2468–2471.

[18] Y. Morvan, D. Farin, and P. H. N. de With, “Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images,” in *Proc. ICIP*, vol. 5. Oct. 2007, pp. 105–108.

[19] P. Merkle, Y. Morvan, A. Smolic, K. Mûller, P. H. N. de With, and T. Wiegand, “The effect of depth compression on multiview rendering quality,” in *Proc. 3DTV Conf.*, May 2008, pp. 245–248.

[20] S. Grewatsch and E. Mûller, “Fast mesh-based coding of depth map sequences for efficient 3-D video reproduction using OpenGL,” in *Proc. Int. Conf. Visualization Imaging Image Process.*, Sep. 2005, pp. 66–71.

[21] S. Y. Kim and Y. S. Ho, “Mesh-based depth coding for 3-D video using hierarchical decomposition of depth maps,” in *Proc. ICIP*, vol. 5. Oct. 2007, pp. 117–120.

[22] K. J. Oh, S. Yea, A. Vetro, and Y. S. Ho, “Depth reconstruction filter for depth coding,” *IET Electron. Lett.*, vol. 45, no. 6, pp. 305–306, Mar. 2009.

[23] L. McMillan and G. Bishop, “Plenoptic modeling: An image-based rendering system,” in *Proc. SIGGRAPH ACM Trans. Graph.*, 1995, pp. 39–46.

[24] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” in *Proc. SIGGRAPH ACM Trans. Graph.*, Aug. 2004, pp. 600–608.

[25] U. Fecker, M. Barkowsky, and A. Kaup, “Improving the prediction efficiency for multi-view video coding using histogram matching,” in *Proc. PCS*, Apr. 2006, pp. 2–16.

[26] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE Int. Conf. Comput. Vision*, Jan. 1998, pp. 839–846.

[27] K. J. Oh, S. Yea, A. Vetro, and Y. S. Ho, “Virtual view synthesis method and self evaluation metrics for free viewpoint television and 3-D video,” *Int. J. Imag. Syst. Technol.*, vol. 20, no. 4, pp. 378–390, Dec. 2010.

[28] *Reference Softwares for Depth Estimation and View Synthesis*, document M15377, ISO/IEC JTC1/SC29/WG11, Apr. 2008.

[29] J. A. Bangham and S. Marshall, “Image and signal processing with mathematical morphology,” *Electron. Commun. Eng. J.*, vol. 10, no. 3, pp. 117–128, Jun. 1998.

[30] A. Telea, “An image inpainting technique based on the fast marching method,” *J. Graph. Tools*, vol. 9, no. 1, pp. 25–36, 2004.

[31] K. J. Oh, S. Yea, and Y. S. Ho, “Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video,” in *Proc. PCS*, May 2009, pp. 233–236.

[32] *MSR 3-D Video Sequences* [Online]. Available: http://www.research.microsoft.com/vision/ImageBasedRealities/3DVideoDownload

[33] *WD 3 Reference Software for MVC (JMVC_3_0)*, document JVT-AC207, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Oct. 2008.

[34] *Hierarchical B Pictures*, document JVT-P014, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Jul. 2005.

[35] *Philips (in Coop with 3D4YOU) Response to New Call for 3DV Test Material: Beergarden*, document M16421, ISO/IEC JTC1/SC29/WG11, Apr. 2009.

[36] *An Excel Add-In for Computing Bjontegaard Metric and Its Evolution*, document VCEG-AE07, ITU-T SG16 Q.6, Jan. 2007.

Kwan-Jung Oh received the B.S. degree in electronic computer engineering from Chonnam University, Gwangju, Korea, in 2002, and the M.S. and Ph.D. degrees in information and communications engineering from the Gwangju Institute of Science and Technology, Gwangju, in 2005 and 2010, respectively.

In 2008, he was an Intern with Mitsubishi Electric Research Laboratories, Cambridge, MA. He joined the Samsung Advanced Institute of Technology, Giheung-gu, Younggin-si Gyunggi-do, Korea, in 2010, where he is currently a Research and Development Staff Member with the Future IT Research Center. His current research interests include digital image and video coding, multiview video coding, 3-D video coding, free viewpoint television, depth video coding, image-based rendering, and realistic broadcasting.

Anthony Vetro (S'92–M'96–SM'04–F'11) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY.

He joined Mitsubishi Electric Research Labs, Cambridge, MA, in 1996, where he is currently a Group Manager responsible for research and standardization on video coding, as well as work on display processing, information security, speech processing, and radar imaging. He has published more than 150 papers in these areas. He has also been an active member of the ISO/IEC and ITU-T standardization committees on video coding for many years, where he has served as an ad-hoc group chair and editor for several projects and specifications. Most recently, he was a key contributor to the multiview video coding extension of the H.264/MPEG-4 AVC standard. He also serves as Vice-Chair of the U.S. delegation to MPEG.

Dr. Vetro is also active in various IEEE conferences, technical committees, and editorial boards. He currently serves on the Editorial Boards of IEEE SIGNAL PROCESSING MAGAZINE and IEEE MULTIMEDIA, and as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE TRANSACTIONS ON IMAGE PROCESSING. He served as the Chair of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and on the steering committees for ICME and the IEEE TRANSACTIONS ON MULTIMEDIA. He served as an Associate Editor for IEEE SIGNAL PROCESSING MAGAZINE from 2006 to 2007, the Conference Chair for ICCE in 2006, the Tutorials Chair for ICME in 2006, and as a member of the Publications Committee of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS from 2002 to 2008. He is a member of the Technical Committees on Visual Signal Processing and Communications, and Multimedia Systems and Applications of the IEEE Circuits and Systems Society. He has also received several awards for his work on transcoding, including the 2003 IEEE Circuits and Systems CSVT Transactions Best Paper Award.

Yo-Sung Ho (M'81–SM'06) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990.

He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, NY, where he was involved in the development of the advanced digital high-definition television system. In 1993, he rejoined the technical staff of ETRI and was involved in the development of the Korea direct broadcast satellite digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, where he is currently a Professor with the Information and Communications Department. Since August 2003, he has been the Director of Realistic Broadcasting Research Center at GIST. His current research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3-D television, and realistic broadcasting.

Dr. Ho is currently serving as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS VIDEO TECHNOLOGY. He gave several tutorial lectures at various international conferences, including the IEEE International Conference on Image Processing in 2009 and 2010, the IEEE International Conference on Multimedia and Expo in 2010, and the Pacific-Rim Conference on Multimedia in 2006 and 2008.