

## PAPER

## Adaptive Depth-Map Coding for 3D-Video

Kyung-Yong KIM<sup>†</sup>, Gwang-Hoon PARK<sup>†a)</sup>, and Doug-Young SUH<sup>†</sup>, Members

**SUMMARY** This paper proposes an efficient adaptive depth-map coding scheme for generating virtual-view images in 3D-video. Virtual-view images can be generated by view-interpolation based on the decoded depth-map of the image. The proposed depth-map coding scheme is designed to have a new gray-coding-based bit-plane coding method for efficiently coding the depth-map images on the object-boundary areas, as well as the conventional DCT-based coding scheme (H.264/AVC) for efficiently coding the inside area images of the objects or the background depth-map images. Simulation results show that the proposed coding scheme, in comparison with the H.264/AVC coding scheme, improves the BD-rate savings 6.77%–10.28% and the BD-PSNR gains 0.42 dB–0.68 dB. It also improves the subjective picture quality of synthesized virtual-view images using decoded depth-maps.

**key words:** 3D-video coding, depth-map coding, free view-point TV

## 1. Introduction

Use of high-definition broadcast service above the HD-level has recently spread widely, aided by rapid progress in multimedia processing technology. Furthermore, there is increasing interest in 3D-TV broadcasting using 3D contents to provide increased realism. 3D-TV broadcasting is expected to gain even greater attention in the next-generation broadcasting service market. The Joint Video Team (JVT) of MPEG (ISO/IEC/JTC1/SC29/WG11) and VCEG (ITU-T/SG16/Q.6) completed the Multi-view Video Coding (H.264/AVC Amendment 4 MVC) standards for efficient video compression of multi-viewed video inputs from more than two cameras [1]. Moreover, MPEG is now planning to introduce its 3D-Video standard [2]. 3D-Video will be a standard that supports various user-produced virtual-view images. Virtual-view images can be infinitely generated by view-interpolation using a depth-map that is the distance between the camera and the real-world object. Therefore, 3D-Video can save the storage capacity since it only needs to transfer a few real-view image data and their depth-map. The experts in 3D-Video at MPEG are currently concentrating their research on a method to generate depth-maps, as well as a method to generate virtual-view images using depth-maps. They will eventually concentrate on an efficient depth-map coding scheme, since there are currently no depth-map-related coding standards. Depth-maps should be compressed with real-view image inputs, and then

transmitted through the transmission media or stored. The best existing depth-map compression scheme uses the DCT-based coding method designed for as-is natural images, for example, the H.264/AVC video coding scheme [3]. However, this method does not consider the image characteristics of depth-maps. Research into efficient compression methods that consider depth-map characteristics is needed, since the subjective quality of the synthesized virtual-view image depends upon the accuracy of the depth-map.

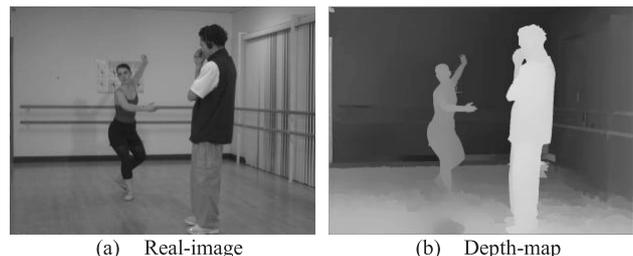
This paper is configured as follows: Sect. 2 points out the problem created by applying DCT-based coding to compressing depth-maps used in 3D-Video. In Sect. 3, the image characteristics of the depth-maps are analyzed in bit plane-by-bit plane fashion. Then an adaptive depth-map coding scheme is proposed. Section 4 shows the coding performance of the proposed method in comparison with existing state-of-the-art DCT-based coding schemes such as H.264/AVC. Section 5 concludes this paper.

## 2. Depth-Map Coding

A depth-map used in 3D-Video is defined as the expression of the corresponding pixel's depth information in a certain number of bits, using the same resolution with the distance between the camera and the real-world object.

An example depth-map in Fig. 1 shows the Ballet image—an MVC test sequence, and its depth-map. The actual depth-map in Fig. 1 is the depth information expressed in 8-bit data that corresponds to each pixel. The closer to the camera a pixel is, the bigger (brighter) the value is, and vice versa.

Since a depth-map image is an expression of the distance between the camera and the actual object, unlike a natural-image, its characteristics are so smooth in both inner parts of the objects and background, together with the cliff-



**Fig. 1** Real-image and its depth-map in Ballet sequence.

Manuscript received October 6, 2009.

Manuscript revised February 22, 2010.

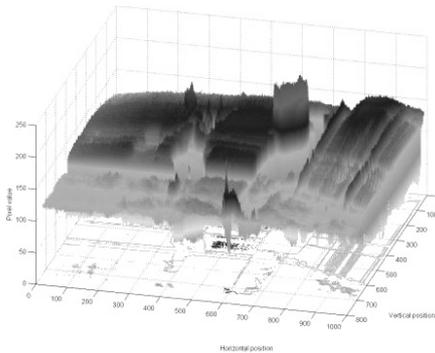
<sup>†</sup>The authors are with Media Lab. (National Research Lab in Korea), College of Electronics and Information, Kyung Hee University, 1, Seocheon, Kiheung, Yongin, Kyunggi, 449–701, Korea.

a) E-mail: ghpark@khu.ac.kr

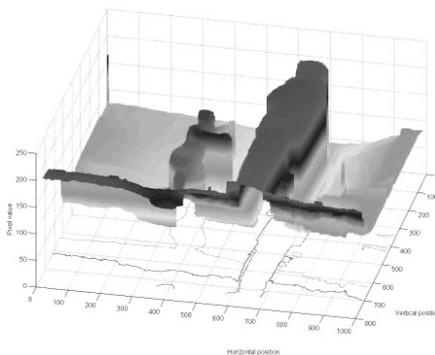
DOI: 10.1587/transinf.E93.D.2262

like edges on object-boundaries as shown in Fig. 2. Moreover, every bit-plane image looks simple when the depth-map images are separated into several bit planes.

At present, a DCT (Discrete Cosine Transform)-based coding scheme, such as H.264/AVC, is generally used as a depth-map coding method [4]. An international standard for depth-map coding methods has not yet been defined. Only a coding standard for the depth-map related parameters (front



(a) 3-D graph of the natural-image



(b) 3-D graph of the depth-map

**Fig. 2** 3-D graph represented with each pixel-level of the natural-image and its depth-map in Fig. 1: (a) natural-image, (b) depth-map.



(a) Original depth-map

(b) Decoded depth-map

**Fig. 3** Comparison of original and decoded depth-maps in Ballet sequence.

and rear-side object position information in the picture) exists in the MPEG-C Part-3 [5]. This parameter is needed in rendering virtual images using depth-maps.

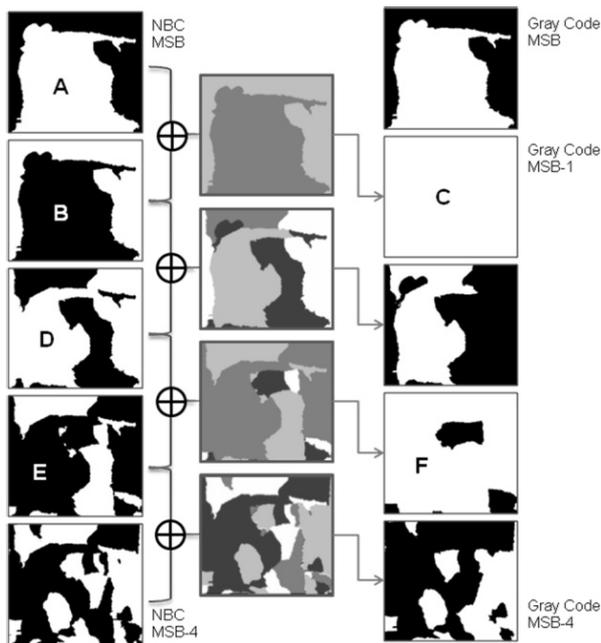
A DCT-based coding scheme is effective when there are high spatial correlations among the neighboring pixels, but it is not effective for an image with big changes between the neighboring pixels, such as in the object-boundary area (cliff-like edges in Fig. 2 (b)). In particular, when DCT-based coding is carried out in a low bit-rate environment, it is difficult to express the object-boundary area accurately due to loss, through quantization, of high frequency components. This causes smudged images as shown in Fig. 3. Figure 3 (a) shows a part of the original depth-map in Ballet sequence (1024 × 768, 15 Hz). Figure 3 (b) shows part of a decoded image compressed by the H.264/AVC coding scheme (with a quantization parameter of 37 for the original depth-map). As seen in Fig. 3 (b), it is confirmed that the images from the decoded depth-map are smudged at the object-boundary area. As mentioned before, DCT-based coding (frequency-domain approach) may produce far amount of bitstreams in order to reconstruct the cliff-like object-boundaries (due to high frequency components), thus follows by obtaining somewhat less coding efficiencies. To improve coding efficiency, this paper introduces an bit-plane based coding method (pixel-domain approach) especially for the object-boundaries of depth-map, and then proposes a combined depth-map coding scheme that can select proper coding method between pixel- and frequency-domain coding approaches that is appropriate to the image characteristics of the depth-map. Gray-coding is applied as a some sort of preprocessing in case the pixel-domain approach is carried out.

### 3. Proposed Depth-Map Coding Scheme

Figure 4 shows bit-plane level image characteristics of depth-map, especially on the object-boundary areas. Left binary images are based on natural binary codes (NBC) and the right images are based on gray-coded images between succeeding bit-plane images.

Gray coding is a particular case of reflected codes. The property of gray-code is that a transition, from a value  $g_M$  to  $g_{M-1}$  or to  $g_{M+1}$ , needs only one bit to be changed. With gray-code representation, many of the bits of a given bit-plane are identical and the data compression techniques can be efficient in the nearly uniform regions [6].

Depth-maps, especially created by segmentation-based depth-map generation schemes [7], mostly have many smooth regions with slow pixel value evolutions and also cliff-like edges on the object-boundaries as shown in Fig. 4. In the segmentation-based depth-map generation scheme, each image is independently segmented, and then an initial disparity for each segment is computed and smoothed using the value of neighboring segments. The smoothing is only performed if the difference of the disparity in the neighboring segments is small. Therefore, there are sharp changes in the object-boundaries of the depth-map [7].



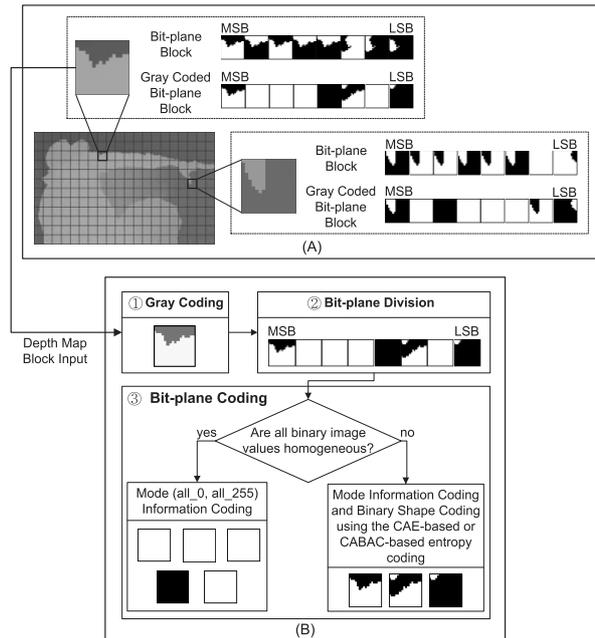
**Fig. 4** Image characteristics of depth-map: Left binary images is based on natural binary codes (NBC). Right binary images are gray coded images between succeeding bit-planes.

Based on the binary image of each bit plane of depth-map, one can easily see the frequent occurrences of completely consistent or reversely consistent image characteristics between bit-plane images. Therefore, binary images can be simplified after the gray coding is applied between bit-planes as shown in Fig. 4. For example, the left images marked as A ( $NBC_{MSB}$ ) and B ( $NBC_{MSB-1}$ ) are completely-reversed, thus the gray-coded image C ( $Gray-coded_{MSB-1}$ ) becomes all 255 (or 1). And also for the binary images of D ( $NBC_{MSB-2}$ ) and E ( $NBC_{MSB-3}$ ), the gray-coded image of F ( $Gray-coded_{MSB-3}$ ) becomes very simplified, and so on.

To use the observed depth-map characteristics to improve the coding efficiency, a gray-coding based bit-plane coding method is proposed in this paper.

The contents in the solid rectangle marked (A) in Fig. 5 is a partial depth-map example of the Breakdancers sequence that belongs to each bit plane after the macroblock ( $16 \times 16$  pixels) of an object-boundary area is separated into several bit planes. As shown in Fig. 5, each block information of the depth-map are converted into the gray codes in case the bit-plane based coding is carried out. When gray coding is applied to the depth-map block of the object-boundary area, the current bit plane's image becomes all zeroes (0) in case the current bit plane's image is completely consistent with the binary image of the previous bit-plane. Furthermore, the current bit plane's image becomes all 255 (or 1) when the current bit plane's image is completely reverse-consistent with the image of the previous bit plane.

Proposed bit-plane based coding method for object-boundary area can be carried out in the following basic



**Fig. 5** Bit-plane analysis of the depth-map in the object boundary block of Breakdancers sequence (A), and proposed bit plane-based coding method for object boundaries (B).

steps, as shown in the block diagram surrounded by the solid rectangle (marked (B) in Fig. 5). Detailed coding method will be explained later.

- Step 1. Each depth-map pixel is gray-coded. In Eq. (1), converting the binary representation of the pixel values in a  $N$ -bit depth-map block, i.e.,  $(a_{N-1}, \dots, a_2, a_1, a_0)_2$ , into the gray code representation  $(g_{N-1}, \dots, g_2, g_1, g_0)_2$  is expressed [8].
- Step 2. 8-bit depth-map is divided into 8 bit planes on a block basis.
- Step 3. Each block of the divided bit planes is repeatedly input to the stage of Bit-plane Coding in the order of MSB, MSB-1, ..., LSB bit planes. Then the bit-stream is finally produced after the coding process is applied. The binary-shape coding scheme of the MPEG-4 Part-2 Visual (ISO/IEC 14496-2) [9] is used as the basis for bit-plane encoding. Entropy coding can be performed by CAE (Context-based Arithmetic Encoding) [10], [11] or the CABAC (Context-Based Adaptive Binary Arithmetic Coding) [12].
- Step 3-1. If all of the binary image values within the block are the same, proceed to Step 3-2, otherwise proceed to Step 3-3.
- Step 3-2. Encoding is only carried out for the mode information, whether the binary image value in the block is all 0(0) or all 255(1).
- Step 3-3. Binary coding is performed to produce the bit-stream of each bit-plane using the CAE-based or CABAC-based entropy codings.

$$g_i = a_i \oplus a_{i+1} \quad 0 \leq i \leq N - 2$$

$$g_{N-1} = a_{N-1} \quad (1)$$

For the object-boundary regions, bit-plane coding method after applying gray coding to the depth-map indicates a higher coding efficiency than that of the DCT-based coding scheme.

On the other hand, the DCT-based coding scheme has higher compression efficiency within the object or at the background area. Therefore, this paper also proposes an adaptive depth-map coding scheme to provide relatively higher compression efficiency, compared to the existing DCT-based coding scheme (H.264/AVC). It thus can produce superior subjective picture quality by using both a bit-plane coding scheme in the object-boundary area and a DCT-based coding scheme either within the object or for the background area. The proposed depth-map coding scheme uses the following two methods to improve compression efficiency.

- (1) A gray-coding-based bit-plane coding method that can utilize the completely consistent or completely reverse-consistent characteristics between the bit-plane data, especially observed in the object-boundary area.
- (2) Adaptive selection between a bit-plane coding method for the data at the object boundary and a DCT-based coding method for depth-map data inside the object or for the background.

The encoder block diagram of the proposed coding scheme is illustrated in Fig. 6. As indicated in Fig. 6, encoding is performed on a certain block size (a 16 ×

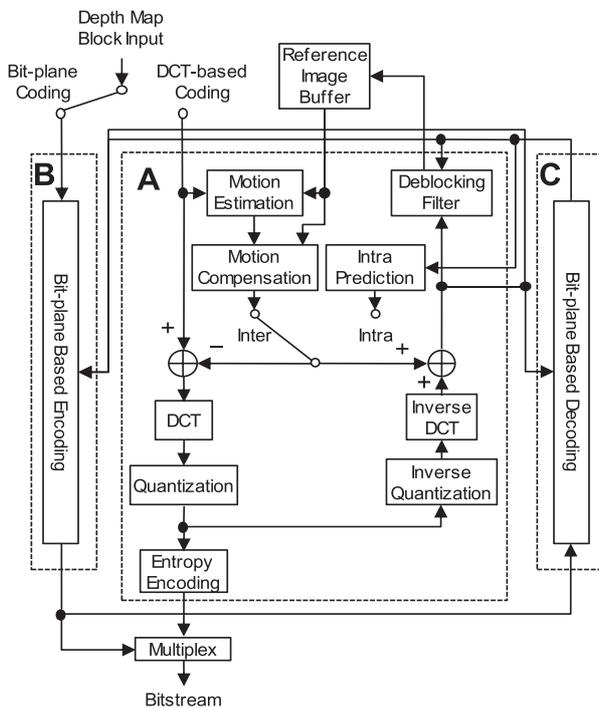


Fig. 6 Encoder structure of the proposed coding scheme.

16 macroblock unit) by separating the input depth-map. For each input block, the best coding mode is selected: either the bit-plane coding mode or the DCT-based coding mode, based on the rate-distortion (RD) cost. Then each block is encoded and transmitted to the decoder with that selected mode's information. The proposed mode selection is illustrated in Fig. 7.

Two cost values are calculated:  $Cost_A$  (coding costs using H.264/AVC DCT-based coding) and  $Cost_B$  (coding costs using gray-coding-based bit-plane coding). Then, the mode that produces lower cost is selected. The RD-optimization technique that is being used in H.264/AVC reference encoder software applied to calculate each cost [13]. For the DCT-based coding, the proposed scheme uses the H.264/AVC encoder, as shown in the dotted box 'A' in Fig. 6. The H.264/AVC video codec performs encoding in either intra mode or inter mode as follows.

- (1) The predicted-block in intra mode is generated for the current block through spatial prediction using the already-coded neighboring blocks. For the inter mode, the predicted block is generated by finding the area best matched with the current block from the reference image by performing motion estimation (ME) and motion compensation (MC) processes.
- (2) The residual block is generated by calculating the differences between the current and predicted blocks.
- (3) Residual-block coding is performed in the following order: transform, quantization, and entropy coding.
- (4) Since H.264/AVC performs inter-frame predictive coding, the current encoded block needs to be decoded and stored as a reference-block image for encoding future image input. Therefore, the quantized block coefficients are transformed into a residual-block image through the inverse quantization and inverse transform processes. This residual-block image is added to the predicted block to generate a

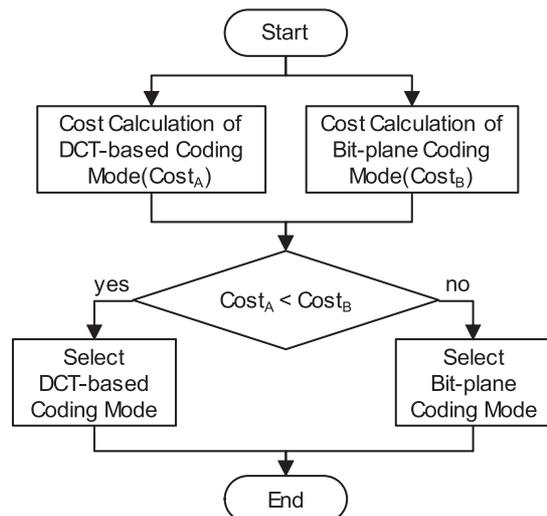


Fig. 7 Proposed method for mode selection.

reconstructed-block image.

- (5) The reconstructed-block image is used as the reference-block image for DCT-based or bit plane-based encoding.

For the bit plane-based encoding (see dotted box 'B' in Fig. 6), the proposed coding scheme uses the encoding structure shown in Fig. 8. The detailed procedure is explained as follows.

- (1) The pixels in each depth-map block are gray-coded.
- (2) The gray-coded N-bit depth-map is divided into N bit planes on a block basis. N is usually 8.
- (3) The rate control process decides the number of bit planes to be encoded according to the quantization parameters (QP) to obtain the desired bit rates, as shown in Fig. 9. If the QP value is less than or equal to 22, the upper 7 bit planes, such as MSB, MSB-1, ..., MSB-6, are encoded. If the QP value is between 23 and 32, the upper 6 bit planes from MSB to MSB-5 can be encoded. If the QP value is above than 32, only 5 bit planes can be encoded to obtain desired bit-rates. The QP parameters of the proposed method are the same values that used in the H.264/AVC coding scheme. For more sensitive rate control, down-sampling based control may be carried out as used in MPEG-4 Part 2 Visual shape coding scheme [9] for each bit-plane. However it may also introduce more

computational complexities in encoding and decoding processes.

- (4) Repetitive bit plane-basis coding, from the MSB bit plane until the bit plane that the rate control process decided, is carried out. The reference bit plane for prediction of current input bit plane is selected using the same level of the bit plane as the bit planes of the gray-coded reference depth-map image.
- (5) The binary-shape coding scheme of the MPEG-4 Part-2 Visual (ISO/IEC 14496-2) [9], [10] is used as the basis for bit-plane encoding. Entropy coding can be performed by the CAE or CABAC. Only the mode information is encoded when all binary image values in the bit-plane block are the same, as shown in Fig. 5. If the binary values in the bit-plane block are not the same, CAE or CABAC are carried out to encode the gray-coded binary data.

As shown in Fig. 6, the proposed coding scheme uses inter-frame predictive coding. Therefore, the current encoded depth-map data has to be decoded and then stored as the reference depth-map image for future encoding.

For bit plane-based decoding, shown in the dotted box C in Fig. 6, the proposed coding scheme uses the decoding structure shown in Fig. 10. The procedure is explained in detail below.

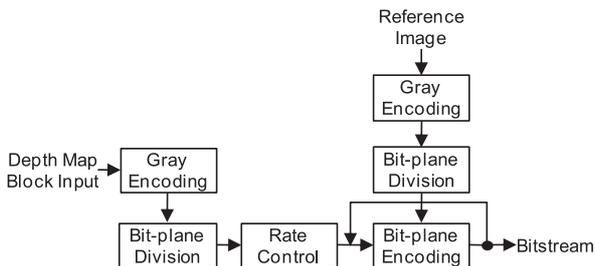


Fig. 8 Encoding structure of the gray-coding-based bit-plane coding method.

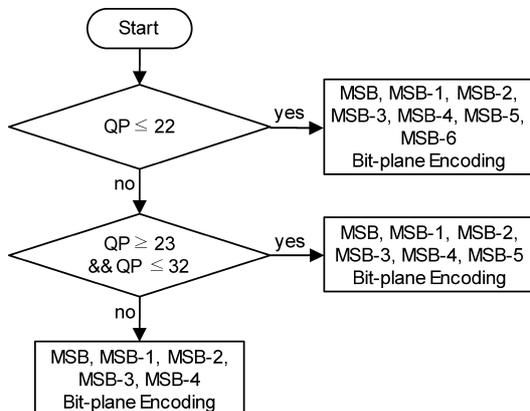


Fig. 9 Bit-rate control according to the quantization parameters.

- (1) Bit-plane decoding is repetitively carried out for the number of encoded bit planes (the number  $m$  should be encoded and then transmitted to the decoder side) to produce the decoded depth-map image. The reference bit plane for the prediction of the current input bit plane is selected using the same bit-plane level as that of the bit planes of the gray-coded reference depth-map image. The binary shape coding scheme of the MPEG-4 Part-2 Visual [9], [10] is used for bit-plane decoding.
- (2) The depth-map image is produced by combining ' $m$ ' bit-plane images.
- (3) By using the inverse gray coding, the gray-coded depth-map image is restored back to the original depth-map type. In Eq. (2), reconverting the gray code representation of m-bit pixels, i.e.,  $(g_{N-1}, \dots, g_2, g_1, g_0)_2$  into the binary representation  $(a_{N-1}, \dots, a_2, a_1, a_0)_2$ , is expressed. The N-bit depth-

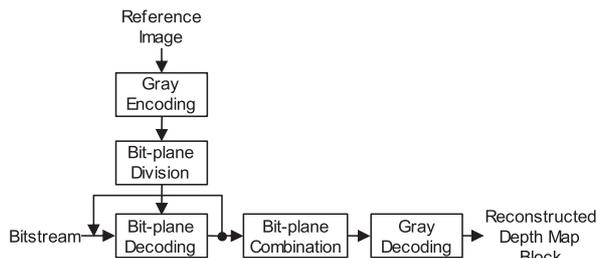


Fig. 10 Decoding structure of the gray-coding-based bit-plane coding.

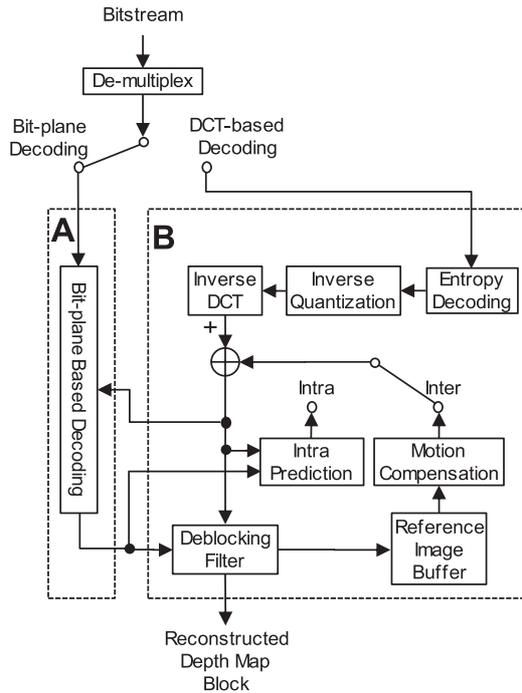


Fig. 11 Decoder structure of the proposed coding scheme.

map is finally reconstructed by applying all 0's from the remaining MSB +  $m$  bit planes, down to the LSB bit plane ( $m < N$ ).

- (4) The reconstructed depth-map is used as the reference image for DCT-based or bit plane-based coding.

$$\begin{aligned}
 a_{N-1} &= g_{N-1} \\
 a_i &= a_{i+1} \oplus g_i \quad 0 \leq i \leq N-2
 \end{aligned}
 \tag{2}$$

The decoding structure of the proposed depth-map coding scheme is illustrated in Fig. 11. The de-multiplexer receives the transmitted bitstream from the encoder, decodes the coding information, and distributes the bitstream according to the corresponding coding mode. Bit plane-based decoding, as shown in Fig. 10, is also carried out, in case the bit-plane decoding mode, as shown in the dotted box A in Fig. 11, is selected. When the DCT-based decoding mode, as shown in the dotted box B in Fig. 11, is selected, a reconstructed depth-map image is produced from the input bitstream using the H.264/AVC decoding scheme.

#### 4. Simulation Results

The proposed depth-map coding scheme has been created by modifying and combining of both H.264/AVC joint-model reference software (JM 13.2) [14] and the shape-coding part of the ISO/IEC 14496 (MPEG-4) Microsoft-FDAM1-2.3-001213 reference software [15]. Two different coding structures have been implemented for proposed depth-map coding scheme, as follows.

- (1) CAVLC-based proposed scheme: DCT-based coding

uses the H.264/AVC with CAVLC, and the bit-plane coding uses the CAE as the entropy coding.

- (2) CABAC-based proposed scheme: DCT-based coding uses the H.264/AVC with CABAC and the bit-plane coding also uses the CABAC as the entropy coding.

Encoding is performed on a certain block size (a  $16 \times 16$  macroblock unit) by separating the input depth-map. For each input block, the best coding mode is selected: either the bit-plane coding mode or the DCT-based coding mode, based on the rate-distortion (RD) cost. Then each block is encoded and transmitted to the decoder with that mode information (1 bits).

In the CAE-based bit-plane coding process, the coding mode (all 0(0) mode, all 255(1) mode, CAE mode) for each bit-plane block is encoded and transmitted to the decoder. The mode information of each bit-plane block is highly correlated due to the characteristics of depth-map. Therefore, mode information of the current bit-plane block is predictively-coded, based on previously encoded neighboring mode information using variable length code (VLC). For the CAE coding mode, the directional information that produce less amount of bitstreams is selected after performing CAE's in both horizontal and vertical directions, and then transmitted to the decoder.

In the CABAC-based bit-plane coding process, all of the coding information of the proposed bit-plane coding also have to be coded using the CABAC. If a coding mode of each bit-plane block is predicted through the coding modes of previously encoded neighboring bit-plane blocks, a flag-bit for correct prediction is transmitted to the decoder. Otherwise, encoding for mode information is carried out. Each binary pixel is encoded in raster scan order, based on its context. Context is constructed by using 3 neighboring pixels that have been previously encoded.

Test video sequences used in this simulation are the 3D-video sequences that are widely used in the JVT MVC and the MPEG 3D-Video standardizations. The depth-maps (hereafter referred to here as "Breakdancers" and "Ballet") provided by Microsoft are also shown in Fig. 12. The coding performance of the proposed scheme is compared with the H.264/AVC video coding scheme (anchor coding scheme). Table 1 gives details of the test conditions for performance measurement.

Figures 13 and 14 show the comparisons of RD-curves that produced by proposed and anchor coding schemes that use the CAVLC for entropy coding in H.264/AVC and the CAE for entropy coding in bit-plane coding for both Breakdancers and Ballet depth-map (view #0) sequences, respectively.

Figures 15 and 16 also show the comparisons of RD-curves that produced by proposed and anchor coding schemes that use the CABAC's for all of the entropy codings for both Breakdancers and Ballet depth-map (view #0) sequences, respectively.

For the Breakdancers test sequence, simulation results are as follows. As shown in Table 2 and Fig. 13, it was

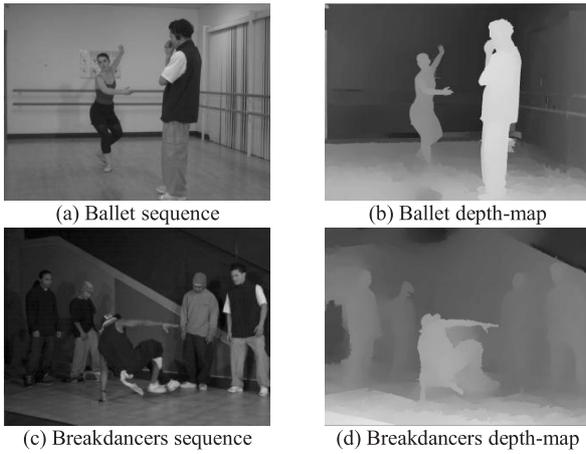


Fig. 12 Test of 3D-video sequences and their depth-maps.

Table 1 Test conditions.

Resolution and frame rate	1024 × 768, 15 Hz
Number of frames	100 Frames
Image format	YUV 4:0:0
Quantization parameter	22, 27, 32, 37, 42
Prediction structure	I-P-P-P-
Entropy coding method	CAVLC, CAE, CABAC

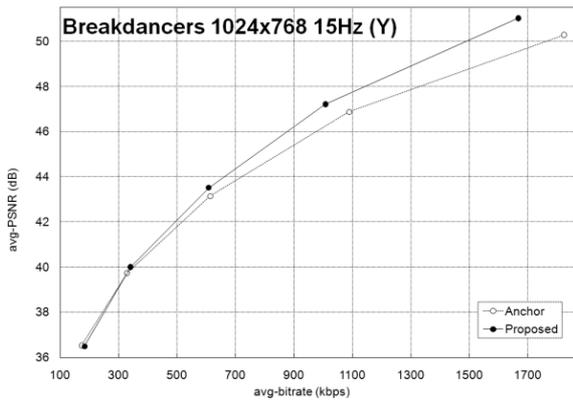


Fig. 13 Comparison of RD-curves that produced by proposed and anchor coding schemes that use the CAVLC for entropy coding in H.264/AVC and the CAE for entropy coding in bit-plane coding for Breakdancers depth-map (view #0) sequence.

confirmed that the coding efficiency of the proposed coding scheme (H.264/AVC: CAVLC, Bit-plane Coding: CAE) is sufficiently superior to that of H.264/AVC (CAVLC) by being about 0.64 dB higher in average BD-PSNR [16], and also by being 9.26 % higher in average BD-rate savings [16]. And as shown in Table 3, and Fig. 15, it was confirmed that the coding efficiency of the proposed scheme (H.264/AVC, Bit-plane Coding: CABAC) is sufficiently superior to that of H.264/AVC (CABAC) by being about 0.54 dB higher in BD-PSNR, and also by being 8.92 % higher in BD-rate savings.

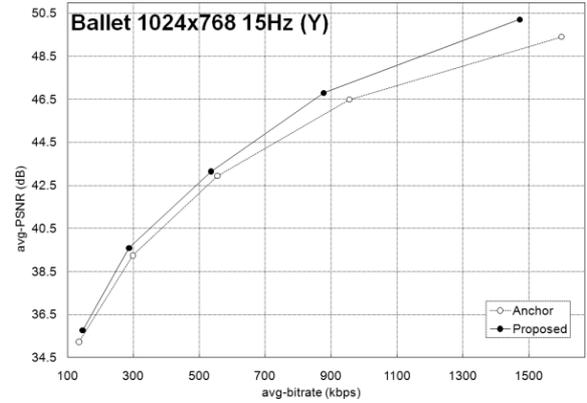


Fig. 14 Comparison of RD-curves that produced by proposed and anchor coding schemes that use the CAVLC for entropy coding in H.264/AVC and the CAE for entropy coding in bit-plane coding for Ballet depth-map (view #0) sequence.

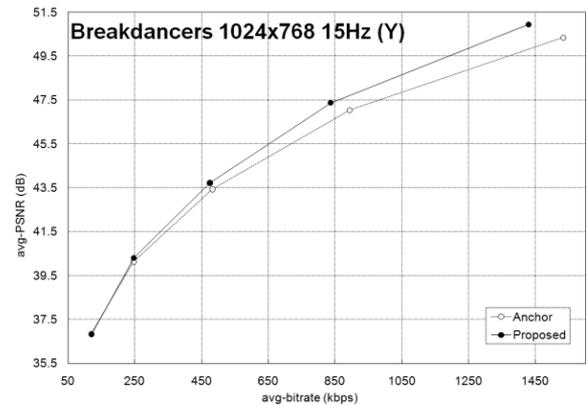


Fig. 15 Comparison of RD-curves that produced by proposed and anchor coding schemes that use the CABAC's for all of the entropy codings for Breakdancers depth-map (view #0) sequence.

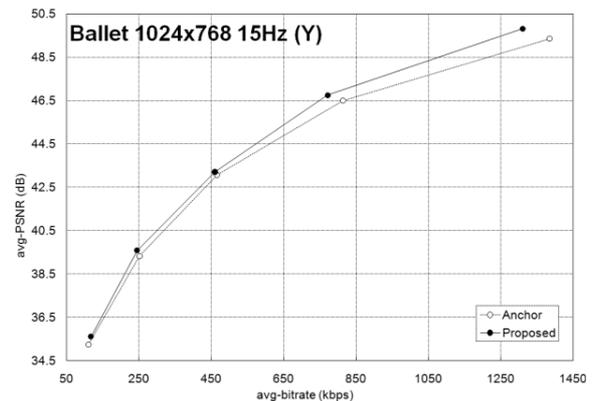


Fig. 16 Comparison of RD-curves that produced by proposed and anchor coding schemes that use the CABAC's for all of the entropy codings for Ballet depth-map (view #0) sequence.

For the Ballet test sequence, simulation results are as follows. As shown in Table 2, and Fig.14, both the performance gain in the BD-PSNR and the bit-rate savings in the BD-rates of the proposed coding scheme

(H.264/AVC: CAVLC, Bit-plane Coding: CAE) were about 0.68 dB and 10.28 % higher, respectively, in comparison with the anchor scheme (CAVLC). As shown in Table 3, and Fig. 16, both the performance gain in the BD-PSNR and the bit-rate savings in the BD-rates of the proposed coding scheme (H.264/AVC, Bit-plane Coding: CABAC) were about 0.42 dB and 6.77 % higher, respectively, in comparison with the anchor coding scheme (CABAC).

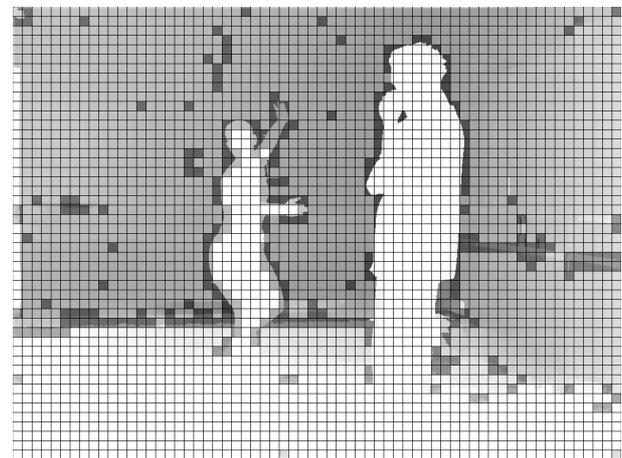
The BD-PSNR's and the BD-rates have been computed using the simulation result with four quantization parameters such as 37, 32, 27, and 22, except 42.

Figure 17 shows the coding mode distributions for tested depth-maps. Dark-gray colored block represents the bit-plane coding is carried out, and also the gray-colored block represents the DCT-based coding is carried out in designated block. As we can easily see, the bit-plane based coding method (pixel-domain approach) is properly used especially for the object-boundaries of depth-map, and also the DCT-based coding method is used for both background and the inner parts of objects. Therefore, it can be recognized that the purpose of our coding scheme is well accomplished in order to improve the coding efficiencies of the depth-maps.

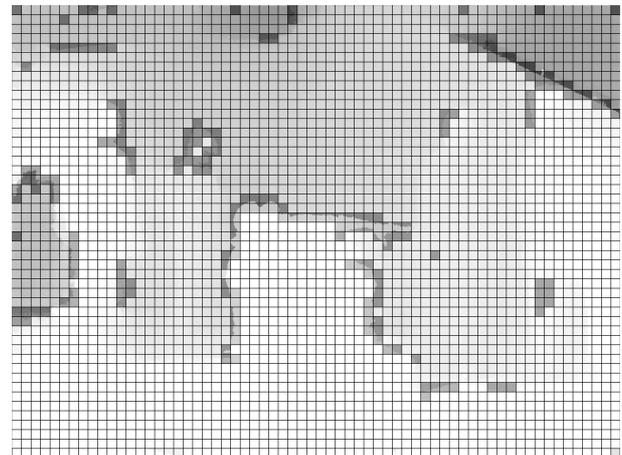
As mentioned before, virtual-view images can be infinitely generated by the view-interpolation method using depth-maps. Its comparison of the subjective picture qualities of the synthesized virtual-view images is explained as follows. In the simulation, virtual-view images were created

by the View Synthesis Reference Software (VSRS 2.0) using decoded depth-maps that are currently being developed for MPEG 3D-Video [17], [18]. Virtual-view images can be created by 3D warping technique [17] using the neighboring actual-view images and their depth-maps. For example, the virtual-view image for view #1, which is corresponding to the actual-view image captured by camera #1, can be created by using the neighboring natural-images of view #0 and view #2 and also their two depth-maps for view #0 and view #2.

Table 4 shows the PSNR result of the synthesized virtual-view image (view #1) that is generated by using two uncompressed depth-maps and also two uncompressed nat-



(a) Coding modes in the Ballet depth-map



(b) Coding modes in the Breakdancers depth-map

**Table 2** Coding efficiency comparisons between CAVLC/CAE-based proposed coding scheme and the CAVLC-based anchor.

	Ballet sequence				Breakdancers sequence			
	Anchor (CAVLC)		Proposed (CAVLC/CAE)		Anchor (CAVLC)		Proposed (CAVLC/CAE)	
QP	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)
42	135.06	35.22	145.68	35.76	174.29	36.53	183.67	36.49
37	298.45	39.24	287.08	39.59	329.28	39.73	341.20	40.00
32	554.52	42.94	535.96	43.15	613.70	43.14	608.34	43.51
27	954.70	46.48	877.39	46.80	1089.00	46.87	1007.76	47.21
22	1598.94	49.40	1472.12	50.21	1824.24	50.28	1667.15	51.02
Proposed vs. Anchor		BD-PSNR (dB)		BD-rates (%)				
Ballet		0.68		-10.28				
Breakdancers		0.64		-9.26				

**Table 3** Coding efficiency comparisons between CABAC-based proposed coding scheme and the CABAC-based anchor.

	Ballet sequence				Breakdancers sequence			
	Anchor (CABAC)		Proposed (CABAC's)		Anchor (CABAC)		Proposed (CABAC's)	
QP	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)	Bitrate (kbps)	PSNR (dB)
42	110.92	35.24	117.71	35.61	120.16	36.84	120.39	36.82
37	298.45	39.24	287.08	39.59	329.28	39.73	341.20	40.00
32	554.52	42.94	535.96	43.15	613.70	43.14	608.34	43.51
27	954.70	46.48	877.39	46.80	1089.00	46.87	1007.76	47.21
22	1598.94	49.40	1472.12	50.21	1824.24	50.28	1667.15	51.02
Proposed vs. Anchor		BD-PSNR (dB)		BD-rates (%)				
Ballet		0.42		-6.77				
Breakdancers		0.54		-8.92				

**Fig. 17** Coding mode distributions for tested depth-maps: dark-gray colored block represents the bit-plane coding is carried out, and also the gray-colored block represents the DCT-based coding is carried out in designated block.

**Table 4** PSNR of the synthesized virtual-view image (view #1) generated by using two uncompressed depth-maps and two uncompressed natural images of both view #0 and view #2.

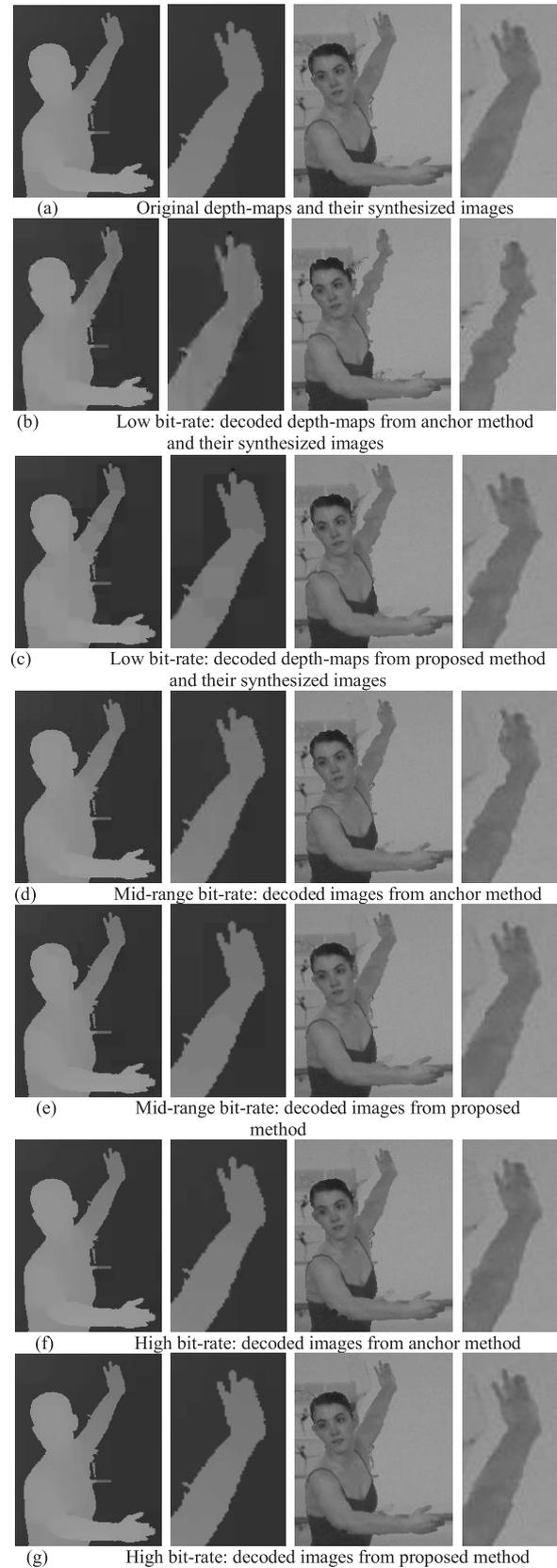
Ballet sequence	Breakdancers sequence
PSNR (dB)	PSNR (dB)
29.98	29.76

ural images of both view #0 and view #2.

Tables 5 and 6 show the PSNR results of the synthesized virtual-view image (view #1) generated by using two uncompressed natural images and two compressed depth-maps of both view #0 and view #2. Table 5 shows the performance comparisons of proposed CAVLC/CAE-based coding schemes with its corresponding anchor, and Table 6 also shows the performance comparisons of two CABAC-based coding schemes.

The PSNR results of the synthesized virtual-view images (view #1), which are created by using the decoded depth-maps with the anchor and proposed coding schemes, are almost the same as shown in Tables 5 and 6. However, it can be easily found that the proposed coding scheme improves the subjective picture qualities of the synthesized virtual-view images than the anchor scheme as follows.

Figure 18 shows the comparisons of the subjective picture qualities of the reconstructed depth-maps (left-hand pair) and their synthesized virtual-view images (right-hand pair) in the Ballet sequence. As can be easily discerned from Fig. 18, the proposed coding scheme is far better than the anchor coding scheme in terms of subjective picture quality comparisons, as explained below. Figure 18(a) shows the original depth-map and its synthesized virtual-view image, which was created by using the original depth-map. As shown in the left-hand pair of images (within Fig. 18(b)), one can easily see the severe image degradations on the decoded depth-map generated by the anchor scheme (H.264/AVC) at a low bitrate, due to the smudges at the object boundary. Therefore, the synthesized virtual-view images created by that decoded depth-map are also widely distorted at the object boundary, as shown in the right-hand pair of images (within Fig. 18(b)). On the other hand, the decoded depth-map using the proposed coding scheme at a low bitrate, as shown in the left pair images



**Table 5** Performance comparisons of CAVLC/CAE-based coding schemes.

	Ballet sequence		Breakdancers sequence	
	Anchor (CAVLC)	Proposed (CAVLC/CAE)	Anchor (CAVLC)	Proposed (CAVLC/CAE)
QP	PSNR (dB)	PSNR(dB)	PSNR(dB)	PSNR(dB)
42	28.85	29.00	28.71	28.71
37	29.27	29.23	28.94	28.88
32	29.57	29.54	29.03	29.00
27	29.74	29.76	29.08	29.05
22	29.81	29.84	29.12	29.12

**Table 6** Performance comparisons of CABAC-based coding schemes.

	Ballet sequence		Breakdancers sequence	
	Anchor (CABAC)	Proposed (CABAC's)	Anchor (CABAC)	Proposed (CABAC's)
QP	PSNR (dB)	PSNR(dB)	PSNR(dB)	PSNR(dB)
42	28.97	28.98	28.71	28.71
37	29.21	29.55	28.93	28.91
32	29.52	29.57	29.03	29.03
27	29.76	29.81	29.09	29.09
22	29.84	29.87	29.11	29.14

**Fig. 18** Comparison of subjective picture quality of the reconstructed depth-maps (left-hand pair) and their synthesized view images (right-hand pair) in Ballet sequence.

(within Fig. 18 (c)), has clear object boundaries. Thus, the object boundaries of the synthesized virtual-view image are much improved, as shown in the right-hand pair of images (within Fig. 18 (c)).

As shown in left-hand pair of images (within Fig. 18 (d) and (f)), the decoded depth-maps using the anchor scheme at both mid and high bitrates have almost-clear object boundaries. However, there still exist smudges in those areas, which create subjective degradations in the object-boundary regions of the synthesized virtual-view images, as shown in the right-hand pair of images (within Fig. 18 (d) and (f)).

On the other hand, decoded depth-maps using the proposed coding scheme at both mid and high bitrates, as shown in the left-hand pair of images within both Fig. 18 (e) and (g), have clear object boundaries. Thus, the subjective quality of synthesized virtual-view images is significantly improved, in comparison with those produced by the anchor scheme, as shown in the right-hand pair of images within both Fig. 18 (e) and (g).

## 5. Conclusions

In this paper, we have proposed an efficient depth-map coding scheme to generate numerous virtual-view images of 3D-video. Differing from the general real images, the depth-map has specific image characteristics, so that the bit-plane images are frequently completely consistent or completely reverse-consistent for every single bit in the area of the object's boundaries. Therefore, our adaptive depth-map coding scheme is a combination of methods. One of these is the gray-coding-based bit-plane coding scheme, which can utilize the completely consistent or completely reverse-consistent characteristics between the bit-plane data. The other method is to adaptively select between the bit-plane coding method for data at the object boundary and the DCT-based coding method (H.264/AVC) for depth-map data inside the object or from the background. Simulation results showed that the BD-PSNR's of the proposed coding scheme were improved by 0.42 dB–0.68 dB, and that its bit-rate savings, measured by BD-rates, were reduced by 6.77%–10.28%, in comparison with the H.264/AVC coding scheme. Our proposed coding scheme was confirmed to have excellent coding efficiency. Moreover, it was also confirmed that using a decoded depth-map to synthesize a virtual-view image produces excellent subjective picture quality.

## Acknowledgments

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the NRL Program (ROA-2005-000-10061-0) and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC support program supervised by the IITA (IITA-2009-C1090-0903-0011) and the excellent research scholarship program of Kyung Hee University graduate school.

## References

- [1] ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC 14496-10:200X/FDAM 1 multi-view video coding," N9978, Hannover, Germany, July 2008.
- [2] ISO/IEC JTC1/SC29/WG11, "Applications and requirements of 3D video coding," N10358, Lausanne, Switzerland, Feb. 2009.
- [3] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G.A. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV—A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol.7, no.11, pp.1606–1621, Nov. 2007.
- [4] ISO/IEC 14496-10:2009, "Information technology—Coding of audio-visual objects—Part 10: Advanced video coding," International Organization for Standardization, 2009.
- [5] ISO/IEC 23002-3:2007, "Information technology—MPEG video technologies—Part 3: Representation of auxiliary video and supplemental information," International Organization for Standardization, 2007.
- [6] M. Abdat and M.G. Bellanger, "Combining gray coding and JBIG for lossless image compression," *Proc. IEEE Int. Conf. Image Process*, 1994 (ICIP 94), vol.3, pp.851–855, Nov. 1994.
- [7] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM SIGGRAPH and ACM Trans. Graphics*, Los Angeles, CA, USA, Aug. 2004.
- [8] J.R. Bitner, G. Ehrlich, and E.M. Reingold, "Efficient generation of the binary reflected gray code and its applications," *Commun. ACM*, vol.19, no.9, pp.517–521, Sept. 1976.
- [9] ISO/IEC 14496-2:2004, "Information technology—Coding of audio-visual objects—Part 2: Visual," International Organization for Standardization, 2004.
- [10] N. Brady and F. Bossen, "Shape compression of moving objects using context-based arithmetic encoding," *Signal Process. Image Commun.*, vol.15, no.7, pp.601–617, May 2000.
- [11] N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2D shape sequences," *Special Session on Shape Coding (ICIP '97)*, vol.1, pp.29–32, 1997.
- [12] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.7, pp.620–636, July 2003.
- [13] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.7, pp.688–703, July 2003.
- [14] F. Heinrich-Hertz-Institut, H.264 Reference Software Version JM13.2, <http://iphome.hhi.de/suehring/html>, May 2008.
- [15] ISO/IEC 14496-5:2001, "Information technology: Coding of audio-visual objects, Part 5: Reference Software," International Organization for Standardization, 2001.
- [16] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T SG16 Q.6, VCEG-M33*, Texas, USA, April 2001.
- [17] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Image Commun.*, vol.24, no.1-2, pp.65–72, Jan. 2009.
- [18] M. Tanimoto and T. Fujii, "View synthesis algorithm in view synthesis reference software 2.0 (VSRS 2.0)," *ISO/IEC JTC 1/SC29/WG11 M16090*, Lausanne, Switzerland, Feb. 2009.



**Kyung-Yong Kim** received the B.S. and M.S. degrees in Computer Engineering from Kyunghee University, Suwon, Korea in 2007 and 2009, respectively. Since 2009, he stayed in Media Lab to study moving picture coding. His current research interests are 3D video coding.



**Gwang-Hoon Park** received the B.S. and M.S. degrees in the Department of Electronic Engineering from Yonsei University, Seoul, Korea in 1985 and 1987, respectively, and also received the M.S. and Ph.D. degrees in the Department of Electrical Engineering and Applied Physics from Case Western Reserve University, Cleveland, OH, U.S.A. in 1991 and 1995, respectively. He was an Principal Researcher at the Information and Telecommunication R&D Center, Hyundai Electronics Industries Co., Ltd., Korea (1995~1997). He was an Associate Professor of the Department of Computer Science, Yonsei University, Wonju, Korea (1997~2001). Since 2001, He has been a Professor of the Department of Computer Engineering, Kyung Hee University, Korea (2001~Present). He has been the co-inventor of the 30 internationally-registered patents and 68 domestically-registered patents. His current research interests are moving picture coding, video signal processing, multimedia system, pattern recognition, and computational intelligence.



**Doug-Young Suh** received his B.S. degree in Nuclear Engineering from Seoul National University in 1980, and Ph.D. degree in Electrical and Computer Engineering from Georgia Tech. in 1990. In September 1990, he joined Korea Academy of Industry and Technology and conducted research on HDTV until 1992. Since February 1992, he is a professor in College of Electronics and Information in Kyunghee University. His research interests include networked video and video compression. He has been working as a Korean delegate for ISO/IEC MPEG Forum since 1996.