Optical Engineering

SPIEDigitalLibrary.org/oe

Efficient multiview depth video coding using depth synthesis prediction

Cheon Lee
Byeongho Choi
Yo-Sung Ho

Efficient multiview depth video coding using depth synthesis prediction

Cheon Lee
Gwangju Institute of Science and Technology
261 Cheomdan-gwagiro
Buk-gu
Gwangju, 500-712, Republic of Korea
E-mail: leecheon@gist.ac.kr

Byeongho Choi
Korea Electronics Technology Institute
Yatap-dong, Bundang-gu Seongnam
Gyeonggi Province, 463-816 Republic of Korea

Yo-Sung Ho
Gwangju Institute of Science and Technology
261 Cheomdan-gwagiro
Buk-gu
Gwangju, 500-712, Republic of Korea

Abstract. The view synthesis prediction (VSP) method utilizes interview correlations between views by generating an additional reference frame in the multiview video coding. This paper describes a multiview depth video coding scheme that incorporates depth view synthesis and additional prediction modes. In the proposed scheme, we exploit the reconstructed neighboring depth frame to generate an additional reference depth image for the current viewpoint to be coded using the depth image-based-rendering technique. In order to generate high-quality reference depth images, we used pre-processing on depth, depth image warping, and two types of hole filling methods depending on the number of available reference views. After synthesizing the additional depth image, we encode the depth video using the proposed additional prediction modes named VSP modes; those additional modes refer to the synthesized depth image. In particular, the VSP_SKIP mode refers to the co-located block of the synthesized frame without the coding motion vectors and residual data, which gives most of the coding gains. Experimental results demonstrate that the proposed depth view synthesis method provides high-quality depth images for the current view and the proposed VSP modes provide high coding gains, especially on the anchor frames. [DOI: 10.1117/1.3600575]

Subject terms: depth video coding; three-dimensional video coding; depth view synthesis; depth map coding.

Paper 100762R received Sep. 24, 2010; revised manuscript received Mar. 14, 2011; accepted for publication May 25, 2011; published online Jul. 6, 2011.

1 Introduction

The three-dimensional(3D) video provides depth impression of the observed scenery with slight different viewpoints between the left and right eyes, which stimulates the human brain to perceive distance of objects. Due to mature 3D technologies from capturing to display, awareness and interests are rapidly increasing among users.[1–3] A key issue in 3D video technology is how to produce a comfortable 3D scene minimizing visual fatigues. Since most of the visual fatigues are induced by the improper camera baseline, it can be solved by selecting two proper viewpoint images among various viewpoint images. In such application, a sufficient number of viewpoint images should be sent to the 3D displays. However, the huge amount of data due to the multiple views is a serious problem for service; hence, we need to develop an efficient video coding.

In response to such needs and interests, many researchers developed various data formats and coding methods for rendering a 3D scene.[4] Particularly, moving picture experts group (MPEG) and joint video team have developed the multiview video coding (MVC), which compresses multiview videos using high correlations between views.[5] It is the latest coding standard designed for coding the multiview videos efficiently. It employs an interview/temporal prediction structure based on the hierarchical B-picture coding to utilize high correlations among views. For the standardization of MVC, many coding techniques had been proposed such as the prediction structure,[6] view synthesis prediction methods,[7–10] the illumination compensation method,[11] and the motion skip mode.[12]

By finalizing the standardization on MVC in 2008, experts of MPEG discussed the advanced 3D video coding method which supports advanced stereoscopic viewing and autostereoscopic displays. Among various data formats supporting the free-view TV or the 3DTV, the multiview video plus depth (MVD) format is selected for the 3D video system. According to the Plenoptic sampling theory, we can transmit only few viewpoints including depth data composed with a sparse camera arrangement instead of sending a large number of views covering wide field-of-view to the decoder.[13] By reconstructing the depth data at both the encoder and decoder, we generate arbitrary viewpoint images using a view synthesis algorithm, e.g., the depth image-based rendering.[14–16] Because of the use of depth data, there are growing interests in coding techniques that take advantage of the correlation among depth views. The 3D video coding is under development by MPEG with a framework which involves the MVD format.[17]

Since 2008, the standardization for the 3D video coding has started by the MPEG 3DV *ad hoc* group as a second phase of free-viewpoint TV (FTV) works.[18] Experts have set an extended framework which provides the high-quality reconstruction of an arbitrary number of views for advanced stereoscopic processing functionalities and to support autostereoscopic displays.[17,18] Currently, the 3D video system under consideration involves the MVD format to render arbitrary viewpoint images. For instance, instead of transmitting nine viewpoint videos to render them with nineview 3D displays, we can transmit only three viewpoints and

0091-3286/2011/$25.00 © 2011 SPIE

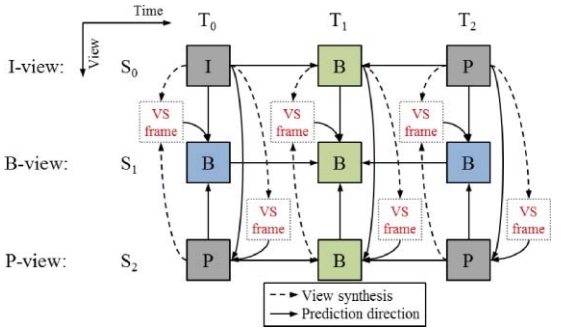


Fig. 1 The prediction structure using the proposed view synthesis prediction.

their corresponding depth videos, and then we generate six intermediate viewpoints using the view synthesis technique. However, the coding issue on depth video coding arises.

The view synthesis prediction (VSP) scheme based on the MVD format is one of the promising technologies for efficient multiview video coding even for depth data. Martinian et al. first proposed the use of the view synthesis prediction as an additional prediction method in which a synthesized picture is generated from neighboring views using depth information and used as a reference for prediction.[7] Developing this method, Yea and Vetro [8] proposed a view synthesis method that employs an optimal mode decision including view synthesis prediction. They added novel prediction modes which infer a depth value and a correction vector.[8] A related scheme by Shimizu et al proposed to encode view-dependent geometry information that enables a view synthesis prediction.[9] Instead of using the available depth map, we have involved a disparity estimation method and a view interpolation method, and then we used additional prediction modes which refer to the generated frame.[10]

We have expanded our previous works and adjusted the algorithm to the depth video coding. Figure 1 describes the basic prediction structure of MVC with GOP (group of pictures) size 2 and the modified one with the proposed method. In depth, in order to compress three views, we encode the first view S_0 without referring to other view; we call this view I-view. After encoding I-view, we encode view S_2 with referring to the reconstructed frames of I-view; we call this view P-view. Finally, we encode view S_1 with referring to both P-view and I-view simultaneously; we call this view B-view. Based on this prediction structure, we can add an additional reference frame generated by a view synthesis method referring to the adjacent reference views; we call this additional frame view synthesis-frame and we named this approach view synthesis prediction. For the P-view case, we synthesize the view synthesis (VS)-frame referring to the reconstructed frame of I-view. For the B-view case, we synthesize the VS-frame referring to both the reconstructed frames of I- and P-view. The view synthesis prediction method utilizes a synthesized image for encoding the current viewpoint video referring to the adjacent reconstructed views. Since the synthesized image has the same time instance and zero disparity to the current frame, we exploit these properties for improving the coding performance. In this paper, based on the MVC prediction structure, we incorporate depth view synthesis using the reconstructed neighboring depth views, and then we employ an additional prediction method named VSP coding.

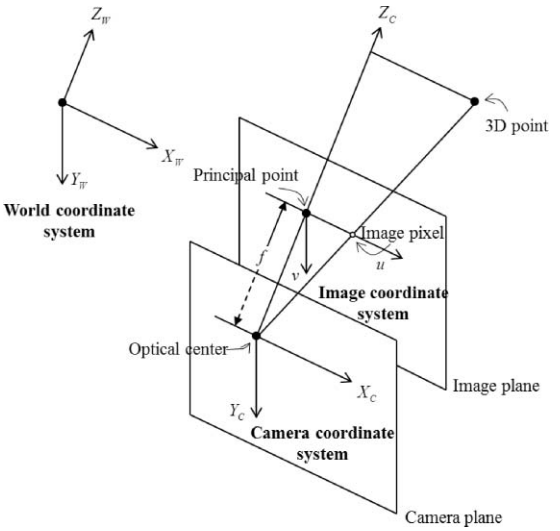


Fig. 2 Three coordinate systems.

The rest of this paper is organized as follows. In Sec. 2, we explain the basics of camera geometry and the 3D image warping technique. In Secs. 3 and 4, we propose a depth view synthesis method and the coding method VSP exploiting the generated depth images, respectively. In Sec. 5, we show experimental results for both synthesized depth images and coding performance using the proposed coding scheme. Concluding remarks are given in Sec. 6.

2 Camera Parameters and Pixel Correspondence

2.1 *Camera Parameters and Pixel Correspondence Between Views*

This section presents the basics of the view synthesis method. To use the depth information of a scene effectively, we need to understand the camera geometry and the multiview geometry.[19] There are three coordinate systems describing the camera itself and the geometrical information of a camera; the world coordinate system, the camera coordinate system, and the image coordinate system as shown in Fig. 2. In a multiview environment, a unique three-dimensional world coordinate system is specified, which is independent from any particular camera. On the other hand, each camera has its own camera system and image coordinate system. The camera coordinate system is three-dimensional with its x_C-y_C plane being the camera plane (also known as principal plane). The optical center of the camera falls within the camera plane. The image coordinate system is two-dimensional in the image plane, where the images are captured. The image plane and camera plane are parallel. The principal point is the intersection point of the optical axis and the image plane.

In order to describe the relationship between different coordinate systems, we define two sets of camera parameters: intrinsic matrix $\mathbf{A}$ and extrinsic matrix $\mathbf{E} = [\mathbf{R}|\mathbf{t}]$. The intrinsic matrix $\mathbf{A}$ represents the transformation from a camera coordinate to its image coordinate. In the world coordinate system, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ describe the direction of the camera and the displacement of the camera from the origin, respectively. When a point $\mathbf{M_w} = [X, Y, Z]^T$ in the world coordinate system is projected to

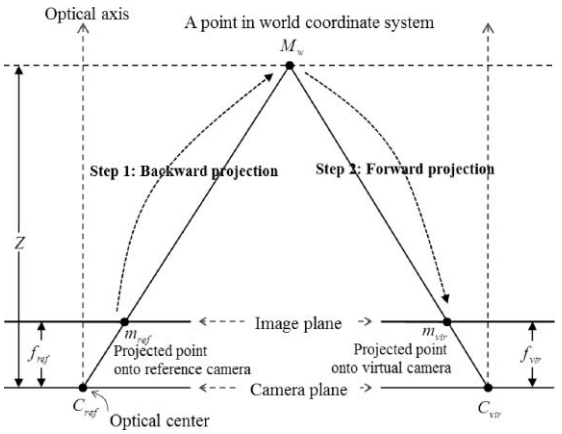


Fig. 3 Finding corresponding pixels between two views.

a pixel $\mathbf{m} = [u\ v]^T$ into the image plane, we can map two points with the following relation as:

$$\tilde{\mathbf{m}} = \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{M}_w + \mathbf{A} \cdot \mathbf{t}, \tag{1}$$

where $\tilde{\mathbf{m}} = [\,u \cdot Z\ v \cdot Z\ Z\,]^T$ represents the projected pixel position in the image coordinate system in homogeneous form.

Let the virtual viewpoint be the target viewpoint to be synthesized; there is neither available color nor depth images. Then we need to find which pixel in the virtual view corresponds to the pixel in the reference view which has both color and depth images. If a point $\mathbf{M}_w$ in the world coordinate system is projected to two separate cameras, as illustrated in Fig. 3, we obtain two pixel positions $\mathbf{m}_{\text{ref}} = [u_{\text{ref}}\ v_{\text{ref}}]^T$ and $\mathbf{m}_{\text{vir}} = [u_{\text{vir}}\ v_{\text{vir}}]^T$ as described in Eq. (2).

$$\begin{cases} \tilde{\mathbf{m}}_{\text{ref}} = \mathbf{A}_{\text{ref}} \cdot \mathbf{R}_{\text{ref}} \cdot \mathbf{M}_w + \mathbf{A}_{\text{ref}} \cdot \mathbf{t}_{\text{ref}} \\ \tilde{\mathbf{m}}_{\text{vir}} = \mathbf{A}_{\text{vir}} \cdot \mathbf{R}_{\text{vir}} \cdot \mathbf{M}_w + \mathbf{A}_{\text{vir}} \cdot \mathbf{t}_{\text{vir}} \end{cases}, \tag{2}$$

where subscriptions ref and vir indicate the reference viewpoint and the virtual viewpoint, respectively. Since the available data are camera parameters of each camera and the depth data of the reference view, we can find 3D points corresponding to pixels of the reference image.

Finding the 3D position from the reference view is called backward projection. From Eq. (1), the projected pixel $\tilde{\mathbf{m}}$ is multiplied by the depth value Z, hence we can rewrite this as $\tilde{\mathbf{m}} = [\mathbf{m}|1]^T \cdot Z$. Using this form deductively, we can find the 3D position $\mathbf{M}_w$ by calculating inverse function of Eq. (2) as:

$$\mathbf{M}_w = \mathbf{R}_{\text{ref}}^{-1} \cdot \mathbf{A}_{\text{ref}}^{-1} \cdot [\mathbf{m}_{\text{ref}}|1]^T \cdot Z_{\text{ref}}(\mathbf{m}_{\text{ref}}) - \mathbf{R}_{\text{ref}}^{-1} \cdot \mathbf{t}_{\text{ref}} \tag{3}$$

where the scalar value $Z_{\text{ref}}(\mathbf{m}_{\text{ref}})$ is the real depth value calculated as:

$$Z_{\text{ref}}(\mathbf{m}_{\text{ref}}) = \frac{1}{\frac{D_{\text{ref}}(\mathbf{m}_{\text{ref}})}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) + \frac{1}{Z_{\text{far}}}}, \tag{4}$$

where $D_{\text{ref}}(\mathbf{m}_{\text{ref}})$ is the pixel value of depth image of reference view. Z_{near} and Z_{far} are the depth range of a physical scene,[20] and those values are given by the sequence provider.

After determining the 3D point $\mathbf{M}_w$, we can re-project it into the virtual view to find $\mathbf{m}_{\text{vir}}$ as described in Eq. (1) as:

$$\tilde{\mathbf{m}}_{\text{vir}} = \mathbf{A}_{\text{vir}} \cdot \mathbf{R}_{\text{vir}} \cdot \mathbf{M}_w + \mathbf{A}_{\text{vir}} \cdot \mathbf{t}_{\text{vir}}, \tag{5}$$

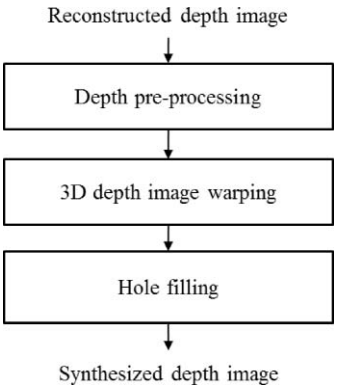


Fig. 4 Steps of depth synthesis.

where $\tilde{\mathbf{m}}_{\text{vir}}$ is a homogeneous representation of a pixel; it is scaled by the depth value of Z_{ref} calculated by Eq. (4), hence the pixel position $\mathbf{m}_{\text{vir}} = [u_{\text{vir}}\ v_{\text{vir}}]^{\text{T}}$ is derived by Eq. (6).

$$\tilde{\mathbf{m}}_{\text{vir}} = Z_{\text{ref}}(\mathbf{m}_{\text{ref}}) \cdot [\mathbf{m}_{\text{vir}}|1]^T = Z_{\text{ref}}(\mathbf{m}_{\text{ref}}) \cdot \begin{pmatrix} u_{\text{vir}} \\ v_{\text{vir}} \\ 1 \end{pmatrix}. \tag{6}$$

Substituting $\mathbf{M}_w$ in Eq. (3) for Eq. (5), we get the relationship between two corresponding pixels as:

$$\begin{aligned} \tilde{\mathbf{m}}_{\text{vir}} = {} & \mathbf{A}_{\text{vir}} \cdot \mathbf{R}_{\text{vir}} \cdot \mathbf{R}_{\text{ref}}^{-1} \cdot \mathbf{A}_{\text{ref}}^{-1} \cdot [\mathbf{m}_{\text{ref}}|1]^T \cdot Z_{\text{ref}}(\mathbf{m}_{\text{ref}}) \\ & - \mathbf{A}_{\text{vir}} \cdot \mathbf{R}_{\text{vir}} \cdot \mathbf{R}_{\text{ref}}^{-1} \cdot \mathbf{t}_{\text{ref}} + \mathbf{A}_{\text{vir}} \cdot \mathbf{t}_{\text{vir}}. \end{aligned} \tag{7}$$

In order to simplify this complex equation, we use a representative equation with $g(\ )$ hereinafter as:

$$\tilde{\mathbf{m}}_{\text{vir}} = g[\mathbf{m}_{\text{ref}}, Z_{\text{ref}}(\mathbf{m}_{\text{ref}})] - \mathbf{t}_{\text{ref,vir}}, \tag{8}$$

where the function $g(\ )$ represents the first term of Eq. (7), and the vector $\mathbf{t}_{\text{ref,vir}}$ replaces the last two terms of Eq. (7) as:

$$\mathbf{t}_{\text{ref,vir}} = \mathbf{A}_{\text{vir}} \cdot \mathbf{R}_{\text{vir}} \cdot \mathbf{R}_{\text{ref}}^{-1} \cdot \mathbf{t}_{\text{ref}} - \mathbf{A}_{\text{vir}} \cdot \mathbf{t}_{\text{vir}}. \tag{9}$$

Using this method, we determine the corresponding pixels between the reference view and the virtual view.

3 Proposed Depth Synthesis Method for 3D Video Coding

In this section, we describe the proposed depth synthesis method for multiview depth video coding. Basically, we use three steps for generating a virtual depth map as described in Fig. 4: depth pre-processing, 3D depth image warping, and hole filling. In the pre-processing step, we reduce erroneous depth values for the reconstructed image using a median filter to minimize image distortions. Next, we find corresponding pixels between the reference view and the virtual view as described in Sec. 2. Finally, we fill the holes referring to the valid pixels either from the synthesized image itself or from the other reference image. The first two steps are the same for both P- and B-view cases, but we use different hole filling methods with respect to the view type.

3.1 *Pre-Processing on Depth Map*

The available depth data at the decoder is the reconstructed one according to the MVC prediction structure. Because the

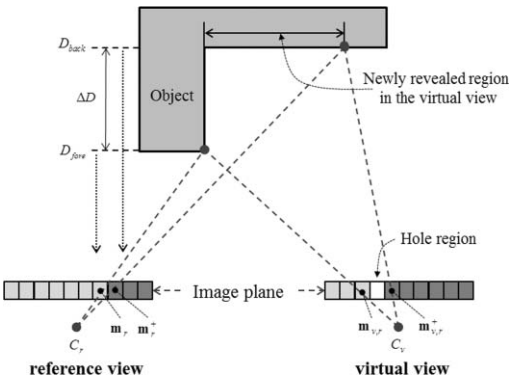


Fig. 5 Hole regions in the virtual viewpoint.

coding method is a lossy compression, the reconstructed data contain erroneous values. Particularly, erroneous depth values induce visual distortions around the object boundaries in the synthesized image. In order to minimize such distortions, we use a window-based median filtering; it is effective for reducing erroneous depth values around abrupt depth changes from the reconstructed depth data.

3.2 *Depth Synthesis Method 1: P-View Case*

In this section, we describe the depth view synthesis method for P-view. The first step is obtaining a viewpoint-shifted depth map using the 3D depth warping. Since P-view can refer to the reconstructed frames of I-view, we define P- and I-view as a virtual view and a reference view, respectively. Let $D_r(\mathbf{m}_r)$ be a depth value with respect to the pixel $\mathbf{m}_r$ of the reference view, and $D_v(\mathbf{m}_v)$ be the depth value with respect to the pixel $\mathbf{m}_v$ of the virtual view. Then, we find the relationship between two pixels as:

$$\begin{cases} D_{v,r}(\mathbf{m}_{v,r}) = \alpha_{v,r}(\mathbf{m}_{v,r}) \cdot D_r(\mathbf{m}_r) \\ m_{v,r} = g[\mathbf{m}_r, Z_r(\mathbf{m}_r)] - \mathbf{t}_{v,r} \end{cases}, \tag{10}$$

where $Z_r(\mathbf{m}_r)$ is the real depth value obtained by Eq. (4), and the scalar value of $\alpha_{v,r}$ describes the visibility of a pixel $\mathbf{m}_{v,r}$ in the virtual view; if it is visible in virtual view, it is set to 1, otherwise, it is set to 0. The set of alpha values is called alpha map. The translation vector $\mathbf{t}_{v,r}$ is the substituted translation vector as:

$$\mathbf{t}_{r,v} = \mathbf{A}_v \cdot \mathbf{R}_v \cdot \mathbf{R}_r^{-1} \cdot \mathbf{t}_r - \mathbf{A}_v \cdot \mathbf{t}_c. \tag{11}$$

When we synthesize a depth image using the 3D warping, holes are generated due to viewpoint changing. As illustrated in Fig. 5, some regions around the foreground object are revealed at the other viewpoint. In detail, if pixels $\mathbf{m}_r$ and $\mathbf{m}^+{}_r$ indicate the foreground object with the depth value D_{fore} and the background object with the depth value D_{back} in the reference view, respectively, the corresponding pixels $\mathbf{m}_{v,r}$ and $\mathbf{m}^+{}_{v,r}$ in the target view are determined by Eq. (10). Since those two adjacent pixels have different depth values, the hole region at the virtual view reveals between two mapped pixels.

In order to fill those holes efficiently, we classified them into two categories. If the width of the hole is wider than one pixel, we call this hole large hole. Otherwise, we call it small hole. The large holes are revealed beside the foreground object at most cases, and the small hole are revealed through the entire depth image in the virtual viewpoint image. For

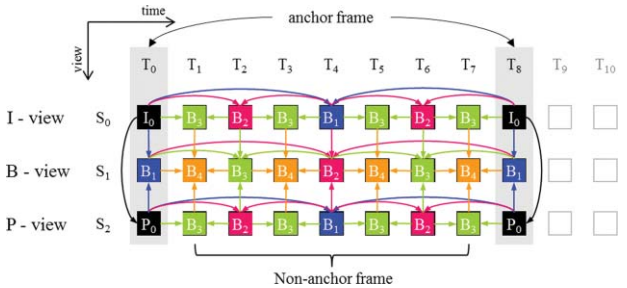


Fig. 6 Basic prediction structure of MVC in 3-view configuration.

the small holes, we fill them using the median filter; we used the 3×3 mask. To fill the large holes, we use the surrounded background depth values, since most likely the true depth values are very close to them.

3.3 *Depth Synthesis Method 2: B-View Case*

Since B-view can refer to two reconstructed frames of I- and P-view simultaneously, we define B-view as a virtual view and I- and P-view as a left view and a right view, respectively. For the B-view case, we use a similar depth warping method with P-view case, but the hole filling method is different. Since the frame to be coded of B-view can refer to two adjacent views, we exploit them in the hole filling process. Let $D_{v,L}$, $D_{v,R}$ be the synthesized depth images with holes warped from the left view to the virtual view and from the right view to the virtual view, respectively, then we can rewrite them as:

$$\begin{cases} D_{v,L}(\mathbf{m}_{v,L}) = \alpha_{v,L}(\mathbf{m}_{v,L}) \cdot D_L(\mathbf{m}_L) \\ m_{v,L} = g[\mathbf{m}_L, Z_L(\mathbf{m}_L)] - \mathbf{t}_{v,L} \end{cases}, \tag{12}$$

$$\begin{cases} D_{v,R}(\mathbf{m}_{v,R}) = \alpha_{v,R}(\mathbf{m}_{v,R}) \cdot D_R(\mathbf{m}_R) \\ m_{v,R} = g[\mathbf{m}_R, Z_R(\mathbf{m}_R)] - \mathbf{t}_{v,R} \end{cases}, \tag{13}$$

where the alpha maps $\alpha_{v,L}$ and $\alpha_{v,R}$ represent the visibility of the synthesized depth images; if the alpha value is zero, its pixel is a hole in the synthesized image, and vice versa.

Since the virtual viewpoint is inside two reference views, most of the corresponding hole regions of $D_{v,L}$ exist in the other synthesized depth image $D_{v,R}$. For example, if $a_{v,L}$ is zero in the synthesize depth image, most likely $a_{v,R}$ is one. Hence we can use this property to fill the holes as:

$$\tilde{D}_{v,L}(\mathbf{m}_{v,L}) = \alpha_{v,L}(\mathbf{m}_{v,L}) \cdot D_L(\mathbf{m}_L) + [1 - \alpha_{v,R}(\mathbf{m}_{v,R})] \cdot D_R(\mathbf{m}_R), \tag{14}$$

$$\tilde{D}_{v,R}(\mathbf{m}_{v,R}) = \alpha_{v,R}(\mathbf{m}_{v,R}) \cdot D_R(\mathbf{m}_R) + [1 - \alpha_{v,L}(\mathbf{m}_{v,L})] \cdot D_L(\mathbf{m}_L). \tag{15}$$

If both alpha values are all zero for one co-located pixel, it is a common hole; invisible at both reference views. We fill them with the inpainting method.[21] After filling all holes, we blend those two synthesized depth images into one final depth image using Eq. (16).

$$D_v(\mathbf{m}_v) = \beta \cdot \tilde{D}_{v,L}(\mathbf{m}_v) + (1 - \beta) \cdot \tilde{D}_{v,R}(\mathbf{m}_v), \tag{16}$$

where $\beta = |t_x^R - t_x^v| / |t_x^L - t_x^R|$is the position of the virtual viewpoint. This final synthesized depth image is used for the

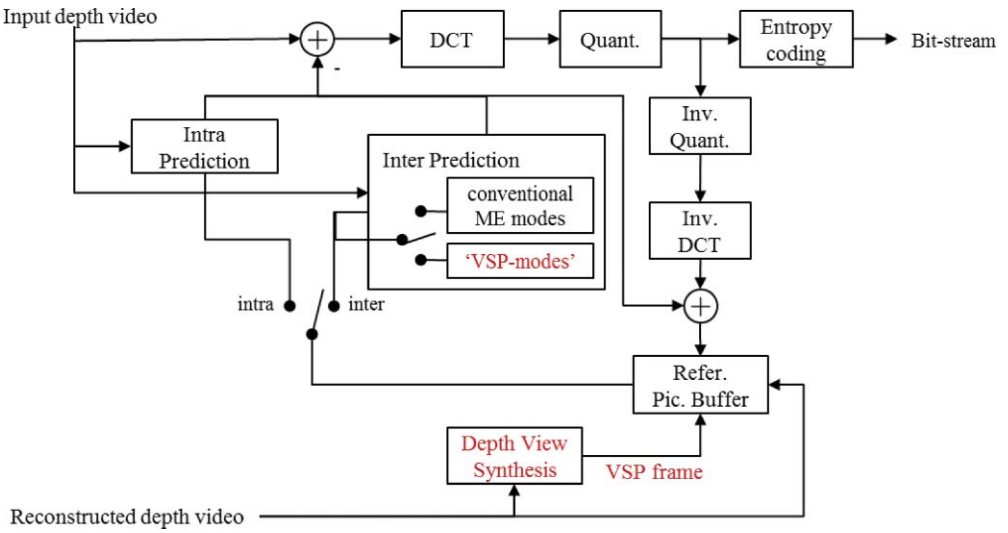
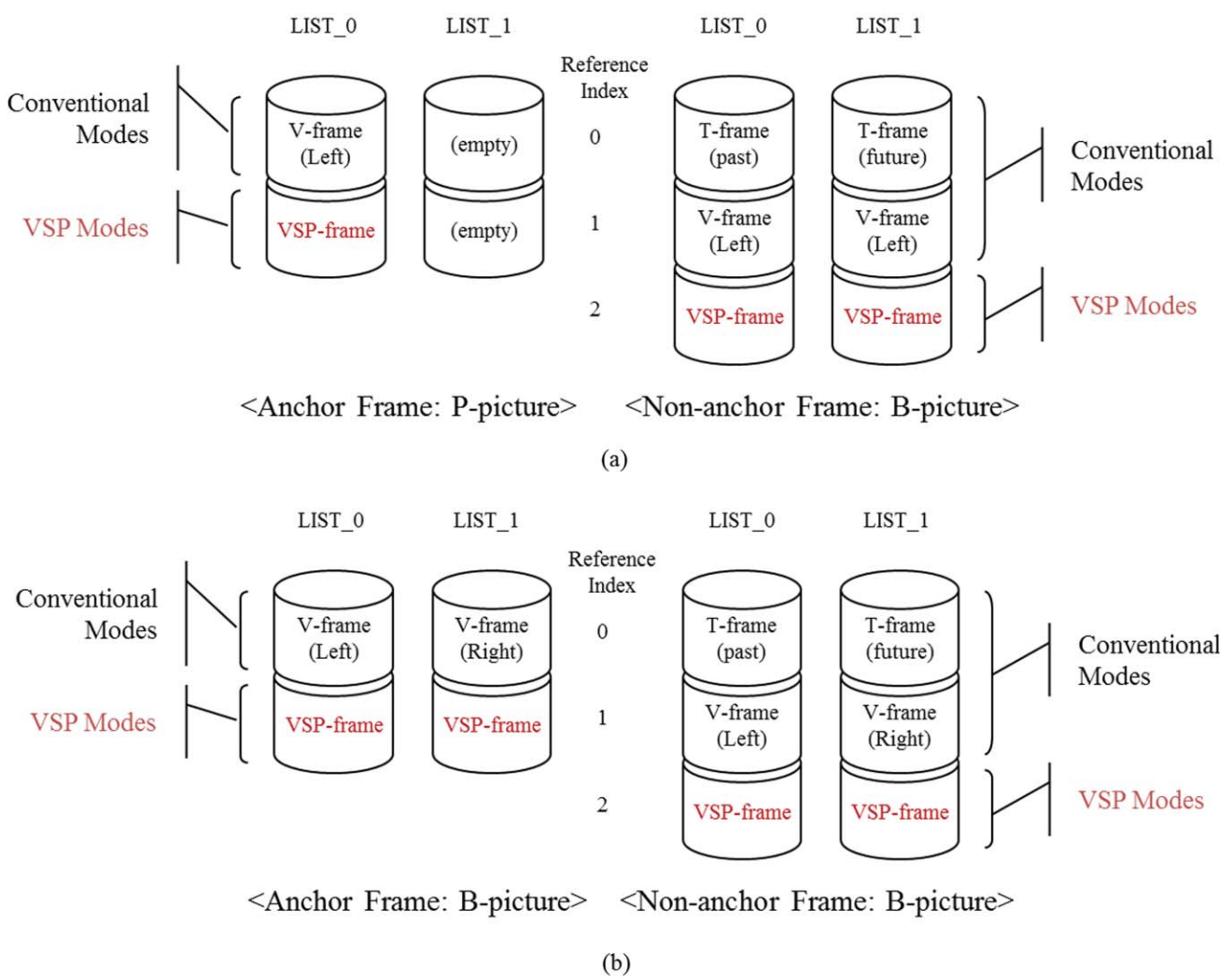


Fig. 7 Structure of the proposed encoder.



(b)

Fig. 8 Reference frame allocations of VSP-frame: (a) P-view case, (b) B-view case.

additional reference frame to the 3D video encoder, which will be described in Sec. 4.

4 Proposed VSP Coding Utilizing Synthesized Depth Image

As we mentioned in Sec. 1, we use the typical MVC prediction structure to encode the multiview depth video; namely the I-B-P prediction structure as shown in Fig. 6, which is a detailed version of the prediction structure of Fig. 1. After encoding I-view, encoding P-view is followed referring to the I-view's reconstructed frames. For example, if the frame at S_2T_0 is the current frame to be coded, it refers to the S_0T_0 frame. If the frame at S_2T_4 is the current frame, it refers to S_0T_4 frame as well as both S_2T_0 and S_2T_8 frames. In the same manner, if the current frame is at S_1T_4 frame, it refers to two S_1T_0 and S_1T_8 frames from the same viewpoint and two S_0T_4 and S_2T_4 frames from the other views; in total, it uses four reference frames.

Based on this prediction structure, we propose a coding method named VSP coding which exploits the interview reference frames effectively. Since the input data represents the depth information for every pixel, we can synthesize any viewpoint depth image using the 3D warping technique. To exploit this, we synthesize a depth map for the current view using the reference frames, and then we involve it as an additionally reference frame at the coder. Since the synthesized depth images have the same time instance with no disparity, it may improve coding performance. However, both increasing coding complexity and additionally the header data for indicating the added frame are inevitable. In this work, we do not focus on the complexity of the coder, but we care for only the coding performance. In the future, we expect that the related hardware technology will overcome this complexity.

Using the depth view synthesis methods as described in Sec. 3, we obtain a depth image for the current view. To exploit it efficiently, we propose the VSP coding method referring to the VSP frame; this is the second contribution of this work. Figure 7 describes the proposed encoder. After generating the depth image, we add it to the reference picture buffer and update the reference lists. In order to use the added VSP frame efficiently, we define additional motion estimation modes named VSP modes, which refers only to the added VSP frame. Detailed descriptions on VSP modes will follow.

4.1 *Updating Reference Lists*

The synthesized depth image using the reconstructed neighboring view is added to the reference picture buffer as shown in Fig. 8. To distinguish the reference frames each other, we defined three representatives as V-frame, T-frame, and VSP-frame. The V-frame indicates the reference frame at the adjacent view, which has the same time instance for the current frame to be coded. The T-frame indicates the reference frame at the same view but different time instance. Since the basic prediction structure is the hierarchical B-picture coding, the T-frame has two frames; one is from the past and the other is from the future.

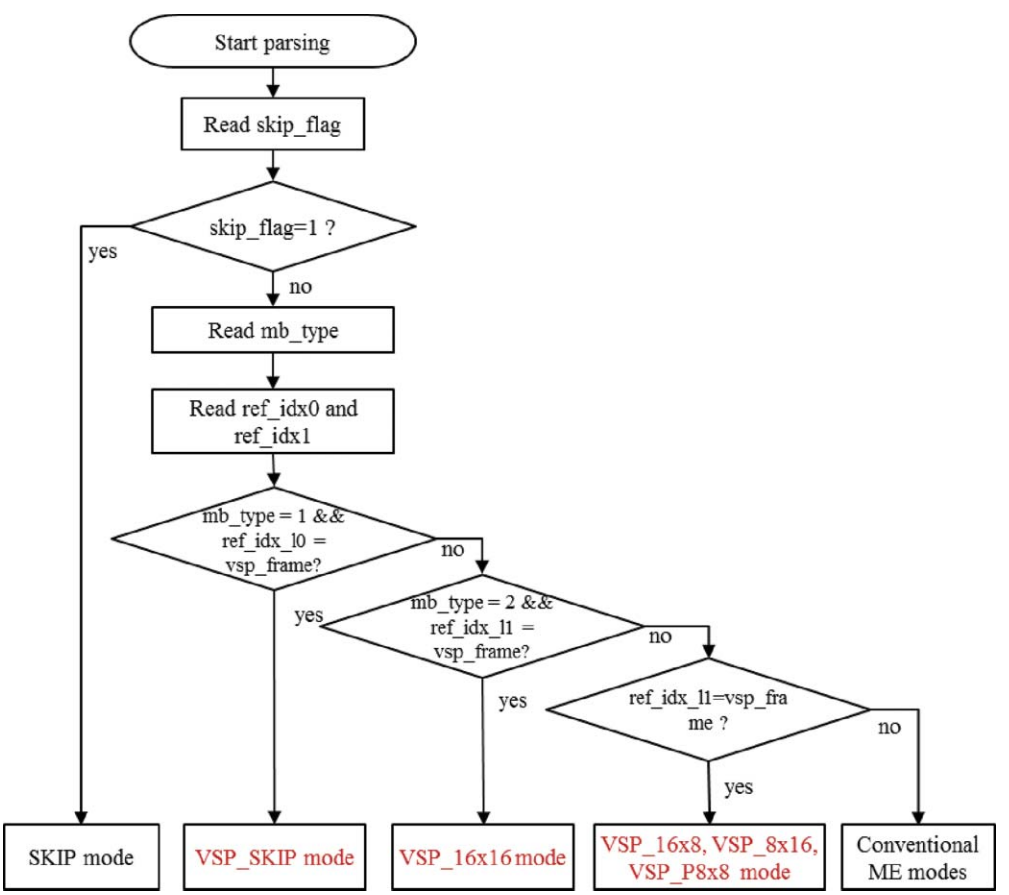


Fig. 9 Parsing process at the decoder.

Table 1 Syntax elements of the VSP modes.

Name of VSP modes	mb_type	Mb Part Pred Mode	Mb Part Width	Mb Part Height	cbp	mvd_l0	mvd_l1	Residual data
VSP_SKIP	1	Pred_L0	16	16	NA	NA	NA	NA
VSP_16×16	2	Pred_L1	16	16	Valid	NA	Valid	Valid
VSP_16×8	6	Pred_L1	16	8	Valid	NA	Valid	Valid
VSP_8×16	7	Pred_L1	8	16	Valid	NA	Valid	Valid
VSP_8×8	22	Pred_L1	8	8	Valid	NA	Valid	Valid

Figure 8(a) describes the allocation method of the reference frames for P-view. According to the view type, we divide them into two cases; one is the anchor frame which refers to only the one V-frame using the conventional motion estimation modes, the other is the nonanchor frame which is located between the anchor frames and refers to both V- and T-frames simultaneously. For the anchor frame of P-view, only LIST_0 is used for prediction, hence the VSP-frame is allocated on the next V-frame as described in Fig. 8 For the nonanchor frame of P-view, an encoder uses both LIST_0 and LIST_1 since it is coded by B-picture coding. The V-frame is allocated on the next T-frames at both lists, and then the VSP-frame is allocated on the next V-frame using the same manner. Different from the P-view case, the anchor frame of B-view refers to two frames from the adjacent views and two frames from the past and future; in total four reference frames are used. The VSP-frame is allocated on the next V-frames for both LIST_0 and LIST_1, respectively. The added VSP-frame is located on the last position of each list. It means that the quality of the synthesized depth image is relatively low compared to the other reconstructed frames.

4.2 *VSP Modes and RD-Optimal Mode Decision*

We use additional prediction modes which are designed for exploiting the VSP-frame. We designed five additional prediction modes which consist of VSP_SKIP, VSP_16×16, VSP_16×8, VSP_8×16, and VSP_P8×8. All modes only refer to the VSP-frame exclusively, i.e., conventional modes refer to the V- and T-frame, as shown in Fig. 9. The key advantage of the VSP-frame is zero disparity frame to the current frame, thus we can copy the co-located block of the VSP-frame as it is; neither motion (or disparity) vector nor residual data. These functionalities are designed at VSP_SKIP mode, which copies all values from the co-located block of the VSP-frame. Different from VSP_SKIP mode, the rest of the VSP modes perform the motion (or disparity) estimation process and encode side information such as cbp (coded-block-pattern), motion (or disparity) vector difference, and residual data.

In order to avoid defining additional syntax elements, we utilized the basic syntax structure of the H.264/AVC. We defined the syntax elements as presented in Table 1. The VSP_SKIP mode uses only LIST_0 prediction with 16×16 block size and mb_type = 1 for the VSP frame. The VSP_16×16 mode uses only LIST_1 prediction with 16×16 block size and mb_type = 2 for the VSP frame. Since both VSP_SKIP and VSP_16×16 modes have the same block size, we differentiated with the prediction directions; this is why we allocated the VSP frame at both lists simultaneously. The rest of the VSP modes perform as described in Table 1. Figure 9 describes the parsing procedure of mode type.

The H.264/AVC involves the rate-distortion optimization method using two cost models: J_{motion} and J_{mode}. The first cost model is used in the motion estimation process to determine the best motion (or disparity) vector having a minimum cost which considers both the difference between the target image and the predicted image and the consuming bits for side information; it can be calculated by:

$$J_{\text{motion}}(\vec{m}, l_m|\text{mb\_type}) = \sum_{X\in\Phi} \left|X - X_p(\vec{m}, l_m)\right| + \lambda \cdot (R_{\vec{m}} + R_{l_m}), \tag{17}$$

where $\vec{m}$ denotes a motion vector per MB with respect to the reference picture index l_m, $R_{\vec{m}}$ and R_{l_m} denote the bits for coding of the motion vector and reference picture index, respectively, and λ is a Lagrange multiplier. X and X_p refer to the pixel values in the target MB X and its prediction, respectively. Since we added five prediction modes, we define a cost model, as shown in Eq. (18), which has to do with the VSP-frame.

$$J_{\text{VSP\_motion}}(\vec{m}, l_{\text{VSP}}|\text{mb\_type}) = \sum_{X\in\Phi} \left|X - X_p^{\text{VSP}}(\vec{m}, l_{\text{VSP}})\right| + \lambda \cdot \left(R_{\vec{m}} + R_{l_{\text{VSP}}}\right), \tag{18}$$

where l_{VSP} denotes the reference picture index of the VSP-frame and $R_{l_{\text{VSP}}}$ denote the bits for coding the reference picture index of it, X_p^{VSP} refers to the predicted pixel values in the VSP-frame. For the VSP_SKIP mode, the rate term $R_{\vec{m}}$ is zero since it uses the co-located block; the motion vector is a zero vector.

After searching motion vectors for each mb_type, the encoder decides which prediction mode is best using the cost criteria J_{mode} as:

$$J_{\text{mode}}(\text{mb\_type}|\lambda_{\text{mode}}) = \sum_{X\in\Phi} (X - X_p)^2 + \lambda_{\text{mode}} \cdot (R_{\text{side}} + R_{\text{res}}), \tag{19}$$

where R_{res} refers to the bits for encoding the residual and R_{side} refers to the bits for encoding all side information including the reference index and the motion vector. Similarly, the cost

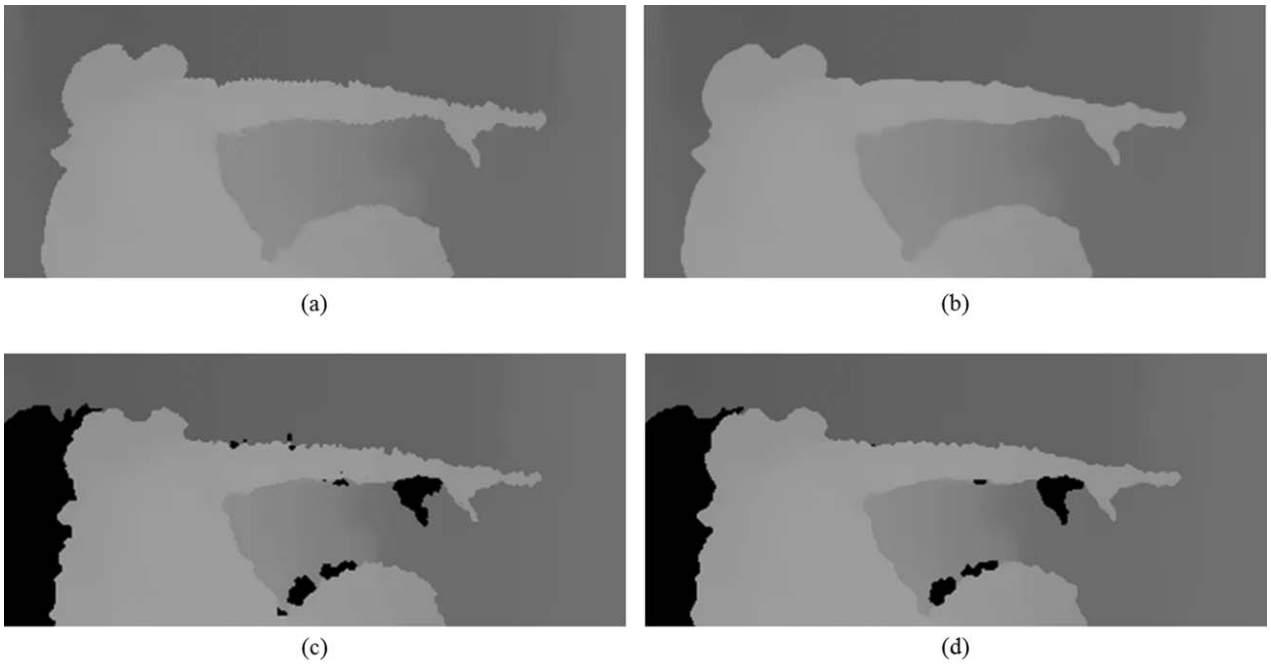


Fig. 10 Warped depth image with pre-processing from view_0 to view_2 of Breakdancers: (a) Reconstructed depth, (b) pre-processed depth, (c) warped depth without pre-processing, (d) warped depth with presxst-processing.

function of VSP modes can be calculated by:

$$J_{\mathrm{VSP\_mode}}(\mathrm{mb\_type}|\lambda_{\mathrm{mode}}) = \sum_{X\in\Phi}\left(X - X_p^{\mathrm{VSP}}\right)^2 + \lambda_{\mathrm{mode}}\cdot(R_{\mathrm{side}} + R_{\mathrm{res}}). \quad (20)$$

Since VSP_SKIP mode excludes coding of the motion vector and residual, we can rewrite the cost function above as:

$$J_{\mathrm{VSP\_SKIPmode}}(\mathrm{mb\_type}|\lambda_{\mathrm{mode}}) = \sum_{X\in\Phi}\left(X - X_p^{\mathrm{VSP}}\right)^2 + \lambda_{\mathrm{mode}}\cdot R_{\mathrm{side}}(\mathrm{mb\_type}, l_{\mathrm{VSP}}). \quad (21)$$

An optimal prediction mode is determined by selecting the minimum cost value including the intra, inter, and VSP prediction modes simultaneously.

5 Experimental Results and Discussion

In this Section, we show experimental results for two contributions: the depth view synthesis and VSP coding methods. The experimental results in this Section are intended to demonstrate the synthesized depth image and the coding performance. Experiments were conducted using three views to make a prediction structure as shown in Fig. 6. We used two types of multiview video sequences with corresponding depth data. The first type of data contains two sequences: "Breakdancers" and "Ballet" (1024×768 @ 15 fps), which are provided by Microsoft Research. Their depth maps have been generated using a stereo matching algorithm.[16] The second type of test data contains two sequences: "Book_arrival" (1024×768 @ 26.67 fps),[22] and "Mobile" (720×540 @ 30 fps),[23] and all depth maps are generated by the depth estimation reference software (DERS 5.0) provided by MPEG 3DV *ad hoc* group.[24]

5.1 *Results on Synthesized Depth Images*

Before synthesizing a depth image for the current frame, we performed the pre-processing on the reconstructed depth image as discussed in Sec 3.1. Figure 10 shows the results of the warped depth images by the pre-processing. The coding errors around the object boundaries as shown in Fig. 10(a) induce distorted synthesized depth values as Fig. 10(c).

Fig. 11 Hole filled synthesized images: (a) P-view case, (b) B-view case.

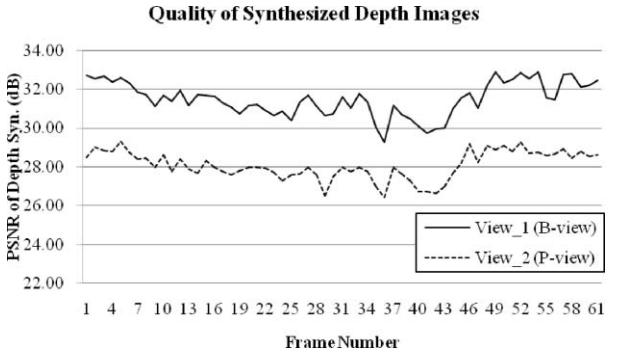


Fig. 12 Quality of the synthesized depth images for Breakdancers coded with QP27.

Unlikely, the median filtering on the reconstructed depth image makes clean object boundaries as shown in Fig. 10(b), and it generates a boundary preserved depth image as shown in Fig. 10(d).

After warping two of the target viewpoints, the hole regions are filled with different methods with respect to the view type, either P- or B-view. In the case of P-view, the hole regions are filled with the valid lower depth value among two neighboring values as shown in Fig. 11(a); it is difficult to recognize which region was the hole region before hole filling. Figure 11(b) is the result of the hole filling on B-view; it shows result better than P-view.

The quality of the synthesized image is affected by the baseline distance between the reference view and the current view. As shown in Fig. 12, the peak signal-to-noise ratio (PSNR) values of the synthesized images of P-view was relatively lower than B-view; the average PSNR values of both P- and B-view were 28.08 and 31.49 dB, respectively. By these results, we can expect that the coding gain will be better at B-view compared to P-view.

5.2 *Results on Multiview Depth Coding Using View Synthesis Prediction*

In this section, we show the results of the depth video coding using the VSP coding method. We tested three view configurations following the MVC prediction structure as shown in Fig. 6. The tested views are selected as the first three views on the MVC sequences and the guided views by the MPEG 3DV *ad hoc* group on the 3DV sequences. All were encoded with a GOP size of 15 for 61 frames in total, and QP sets as 27, 32, 37, and 42. The proposed methods were implemented on the MVC reference software version JMVC 7.0.

Figure 13 and Table 2 describe the coding results compared to the JMVC 7.0 coder. The Bjontegaard Delta bit rate (BDBR) and Bjontegaard Delta PSNR were used to compare the coding performance.[25] The Breakdancers sequence achieved the highest coding gain; the bitrates were reduced as much as -19.82% for B-view. The secondary coding gain was achieved from the Ballet sequence. Those two sequences have high interview correlations; the corresponding pixels of each view are similar each other, hence the synthesized image can provide accurate depth values for the current block. On the contrary, the depth data of 3DV sequences are obtained by depth estimation reference software (DERS), the interview depth correlation is very low, thus the synthesized depth map may be far different from the current frame to be coded.

Another interesting observation on the results is that the coding gains are different according to the view type. The gain of P-view is relatively lower than that of the B-view since the baseline distance is twice further than that of the B-view; the generated hole regions are twice wider than B-view. Also, P-view uses a relatively inaccurate hole filling method to that of the B-view. These are the reasons why the coding gain of P-view is lower than that of the B-view.

A strong advantage of the proposed VSP coding method is the use of additional prediction modes. Since we use the rate distortion (RD) optimization method including the added modes, the coding performance would not be worse. If the VSP frame is not good enough for the coding gain, it would be not selected. On the contrary, if the VSP frame is good enough, VSP modes will be selected; the coding performance will be improved. In this sense, we placed the VSP frame at the last position in each list since it is hard to predict the quality of the VSP frame.

To identify the selected blocks using the VSP modes, we painted with three colors: black, gray, and white. A white block represents the VSP_SKIP mode and a gray block represents the rest VSP modes such as VSP_16×16, VSP_8×16 and so on. The black regions are coded with the conventional modes. Figure 14 describes RD-mode decision maps for VSP modes of the first anchor frame; 0th frame of view 1. Figures 14(a) and 14(b) are the original color and depth images of the first frame, respectively. When we encoded this depth image with four QPs, we obtained four RD-mode decision maps as shown in Figs. 14(c)–14(f). As the QP increases, VSP modes were selected more. Similarly, Fig. 15

Table 2 Coding performance for 3-view configuration.

		Total views (3 views)		B-view (center view)		P-view (rightmost view)	
Test Data	Viewpoints (I-B-P)	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)	BDBR (%)	BDPSNR (dB)
Breakdancers	0-1-2	−11.77	0.60	−19.82	1.11	−16.58	0.86
Ballet	0-1-2	−7.95	0.44	−16.78	1.06	−7.42	0.40
Book_arrival	6-8-10	−2.65	0.15	−7.59	0.45	−0.75	0.05
Mobile	3-4-5	−2.18	0.35	−4.52	1.77	−3.84	0.73
Average		−6.14	0.38	−12.18	1.10	−7.15	0.51

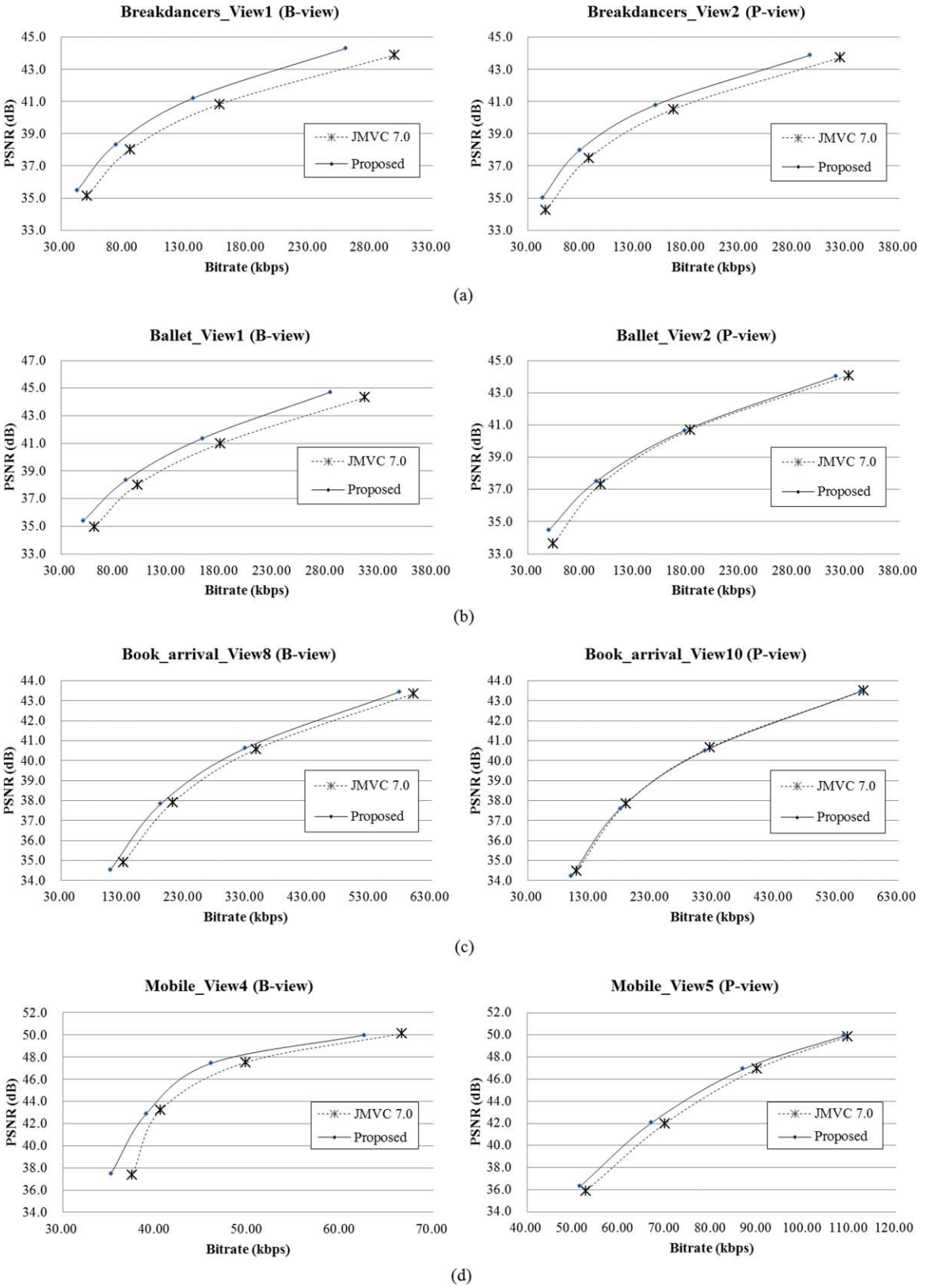


Fig. 13 RD curves of coding results: (a) Breakdancers, (b) Ballet, (c) Book_arrival, (d) Mobile.

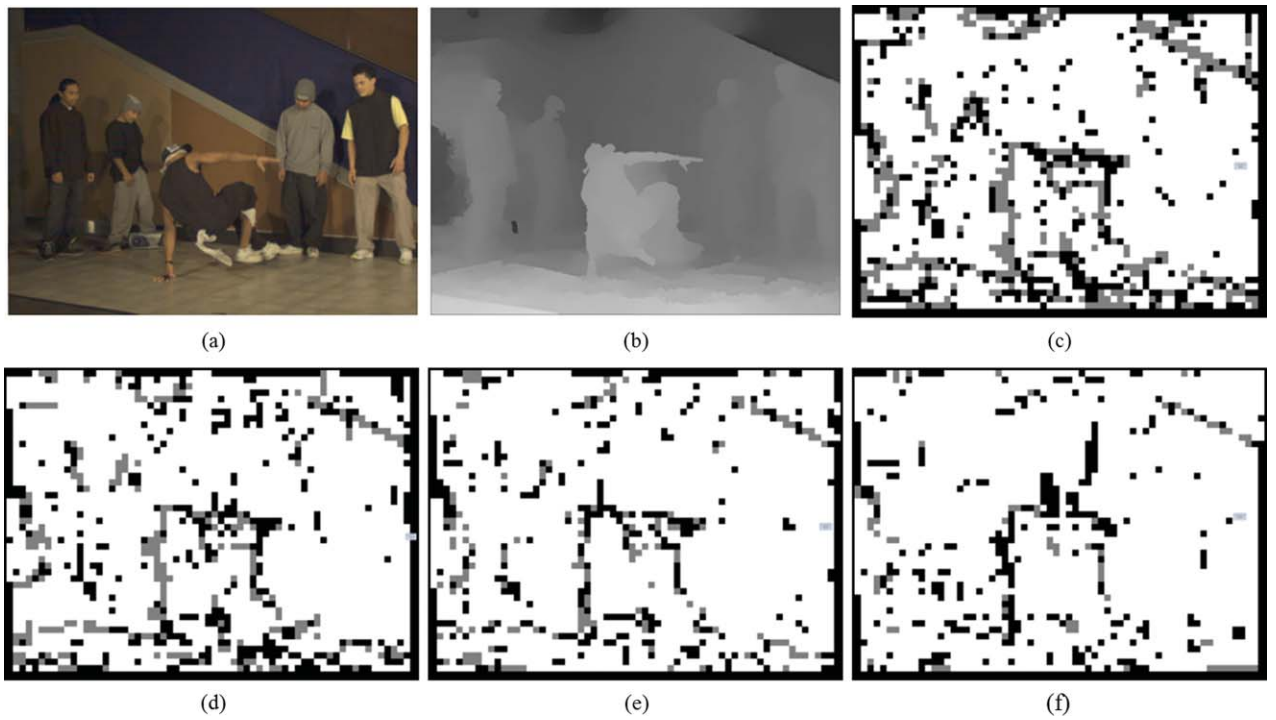


Fig. 14 RD-mode decision maps for anchor frame (0th frame) of Breakdancer. Black: conventional modes, white: VSP_SKIP mode, gray: other VSP-modes: (a) color image, (b) depth image, (c) selected blocks with QP27, (d) selected blocks with QP32, (e) selected blocks with QP37, (f) selected blocks with QP42.

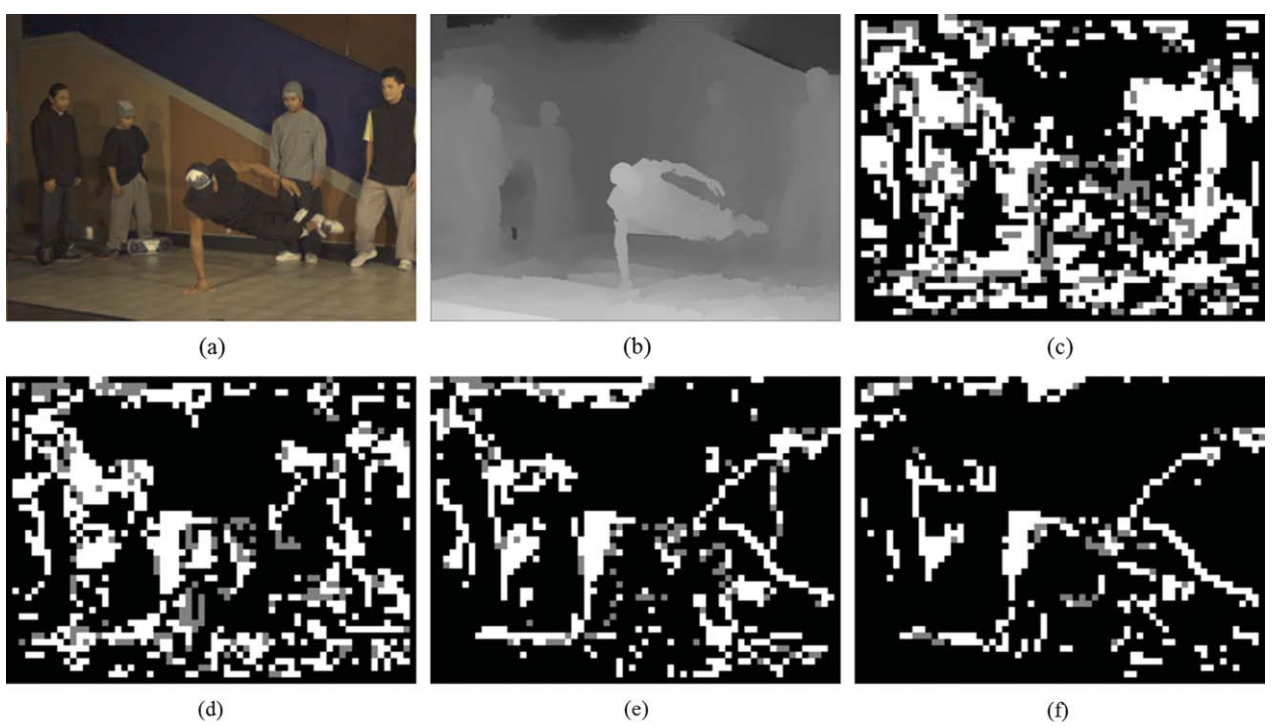
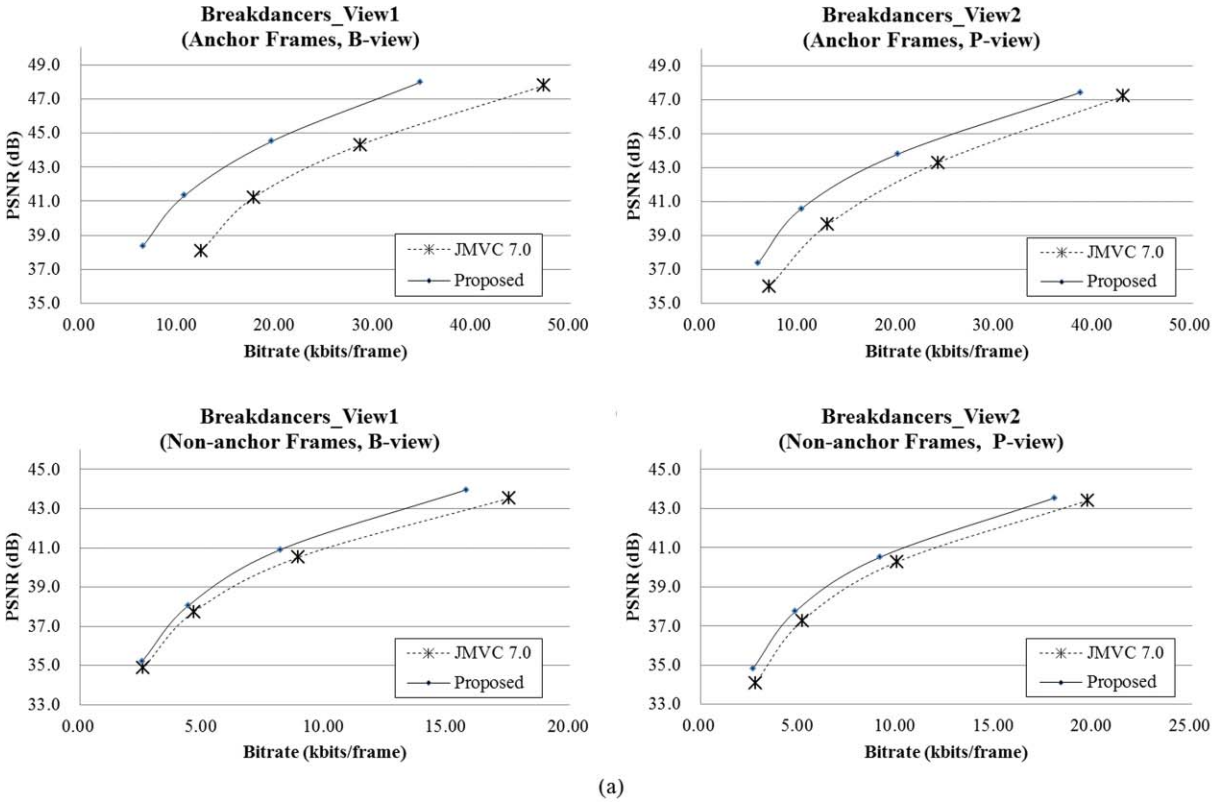
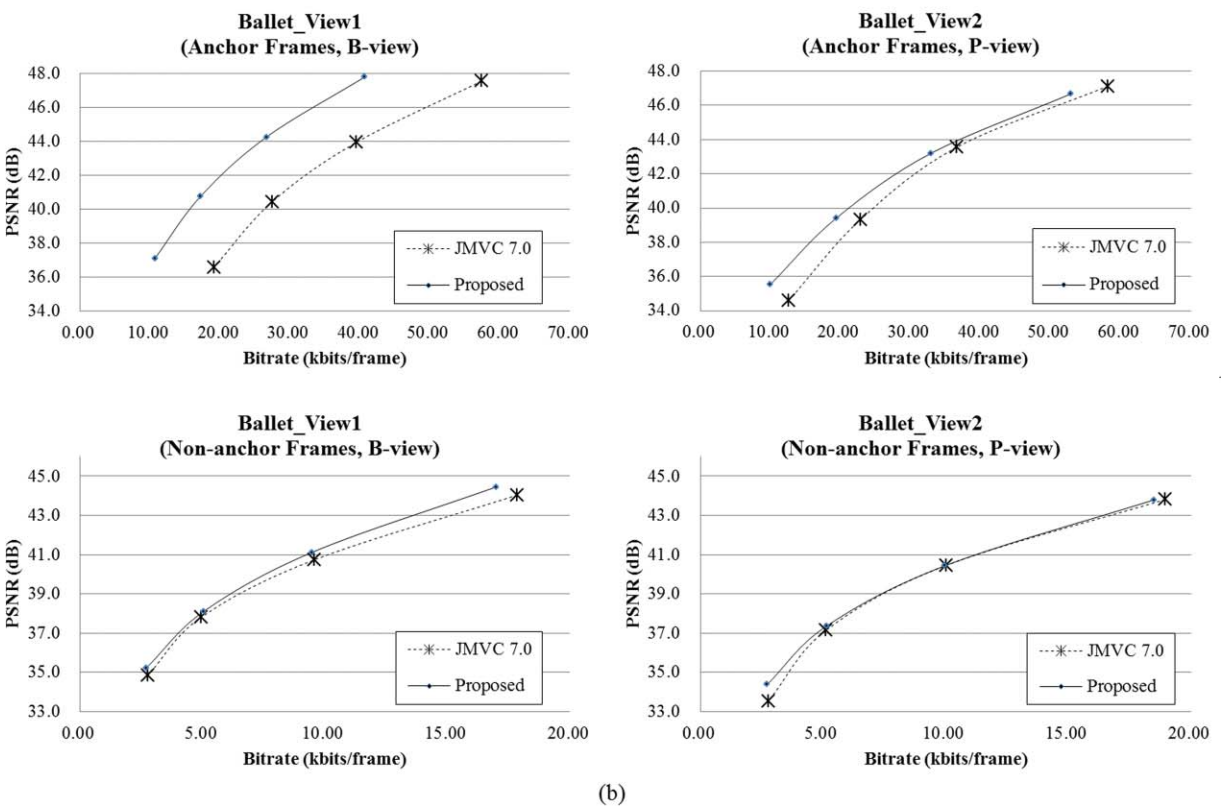


Fig. 15 RD-mode decision maps for nonanchor frame (8th frame) of Breakdancer. Black: conventional modes, white: VSP_SKIP mode, gray: other VSP-modes: (a) color image, (b) depth image, (c) selected blocks with QP27, (d) selected blocks with QP32, (e) selected blocks with QP37, (f) selected blocks with QP42.



(a)



(b)

Fig. 16 RD comparison between anchor and nonanchor frames: (a) Breakdancers, (b) Ballet.

describes the RD-mode decision maps for VSP modes of a nonanchor frame; 8th frame of view 1 at the first GOP. Contrar to the anchor frame case, VSP modes were selected less as QP increases. It means that the temporal prediction with the conventional modes has a higher priority to VSP modes.

Upon this analysis, we noticed that the VSP modes were selected at the anchor frames more; the improvements of coding performance have been achieved by them. To confirm this, we compared the coding gains between the anchor and nonanchor. Figure 16 describes the RD comparison between the two types of frames. Each RD curve was derived by averaging both the bitrates and PSNR values with the number of frames; the *x*-axis represents the average bitrates-per-frame. Definitely, the proposed VSP coding method is effective for the anchor frames.

The proposed VSP coding method was very effective for the depth data since they have high interview correlations between views. If we can synthesize the current viewpoint image using the reconstructed adjacent view correctly, we can use them directly without residual data or motion information. This is why the coding gains are raised at the experiments. However, there are two significant problems on the decoder side. The first problem is a the incensement of the complexity as much as the complexity of the depth synthesis at the decoder. This problem is a critical problem in the sense of manufacturing, but we expect this complexity will be solved by evolutions of technology in the future. Secondary, the proposed VSP coding method employs additional memory for the additional reference frame. Since the proposed method changes the reference buffer management slightly, we think that assigning additional memory is easy to implement in a future 3D video system.

Note that the proposed VSP coding method exploits the syntax structure of MVC to indicate the additional prediction modes such as VSP_SKIP mode; hence no additional bits are included. It means that the coding performance would not drop abruptly even though the quality of the synthesized depth map is poor; we can obtain a safe coding gain.

6 Conclusions

In this paper, we proposed an efficient multiview depth video coding utilizing the adjacent reference views using the view synthesis method. Since the 3D video system involves a multiple viewpoint video, the depth video can use the same prediction structure of the multiview video coding. Therefore, some views can refer to the reconstructed views to improve the coding performance. At first, we synthesize a depth image referring to the adjacent reference frame with the 3D warping method. We use a pre-processing method to reduce an erroneous coding error and two hole filling methods according to the number of reference view points. In order to exploit the synthesized depth image efficiently, we designed five additional prediction modes named VSP modes which refer to the synthesized frame exclusively. By experiments, we confirmed that the coding efficiency have been improved significantly up to 1.11 dB. Most of the gains are achieved from the anchor frames since it has no temporal reference frames. Although the complexity problem still remains, this view synthesis prediction method for depth video coding is very promising for future 3D video systems.

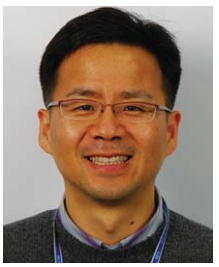
Acknowledgments

This work was supported in part by the IT R&D program of MKE [a development of interactive wide viewing zone SMV optics of 3D display (10035337)], and in part by the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) [NIPA-2011-(C1090–1111-0003)].

References

1. J. Konrad and M. Halle, "3-D Displays and Signal Processing – An Answer to 3-D Ills?," *IEEE Signal Process. Mag.* **24**(6), 97–111 (2007).
2. L. Onural, A. Smolic, and T. Sikora, "An overview of a new European consortium: Integrated three-dimensional television — Capture, transmission and display (3DTV)," presented at the EWIMT, London, (2004).
3. A. Smolic, K. Muller, P. Merkle, C. Fehn, P. Kauff, P. Eiseert, and T. Wiegand, "3-D video and Free Viewpoint Video-Technologies, Applications and MPEG Standards," in *Proc. IEEE ICME*, pp. 2161–2164, Canada (2006).
4. A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems," *IEEE International Conference on Image Processing (ICIP)*, pp. 2448–2451, IEEE, San Diego, CA (2008).
5. G. J. Sullivan, T. Wiegand, and H. Schwarz, Eds., "Editors' Draft Revision to ITU-T Rec. H.264|ISO/IEC 14496 –10 Advanced Video Coding — in preparation for ITU-T SG 16 AAP Consent (in integrated form)," JVT-AD007 (2009).
6. P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient Prediction Structures for Multiview Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, **17**(11), 1461–1473 (2007).
7. E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression," in *Proceedings of the Picture Coding Symposium PCS*, Beijing, China (2006).
8. S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Processing: Image Commun.* **24**(1–2), 89–100, IEEE, EURASIP and IET, Lisbon, Portugal (2009).
9. S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multi-view video coding using 3D warping with depth map," *IEEE Trans. Circuits Syst. Video Technol.* **17**(11), 1485–1495 (2007).
10. C. Lee, K. J. Oh, and Y. S. Ho, "View interpolation prediction for multi-view video coding," *Proc. PCS 2007, Picture Coding Symposium.* (2007).
11. Y. L. Lee, J. H. Hur, Y. K. Lee, K. H. Han, S. Cho, N. Hur, J. Kim, J. H. Kim, P. L. Lai, A. Ortega, Y. Su, P. Yin, and C. Gomila, "CE11: Illumination Compensation," JVT-U052, Hangzhou, China (2006).
12. H. S. Koo, Y. J. Jeon, and B. M. Jeon, "MVC Motion Skip Mode," JVT-W081, San Jose, California (2006).
13. J. X. Chai, X. Tong, S. C. Chan, and H. Y. Shum, "Plenoptic sampling," SIGGRAPH, 307–318 (2000).
14. C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," *Proc. SPIE Stereoscopic Displays and Virtual Reality System XI*, San Jose, CA (2004).
15. Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Processing: Image Commun.* **1–2**, 65–72 (2009).
16. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," SIGGRAPH, **23**(3), 600–608 (2004).
17. ISO/IEC JTC1/SC29/WG11, "Draft report on experimental framework for 3D video coding," MPEG document N11273 (2010).
18. ISO/IEC JTC1/SC29/WG11, "Vision on 3D video," MPEG document N10357 (2009).
19. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press (2004).
20. K. J. Oh and Y. S. Ho, "Non-linear bi-directional prediction for depth coding," *Lect. Notes Comput. Sci.* **5879**, 522–531 (2009).
21. A. Telea, "An image in painting technique based on the fast marching method," *J. Graphics Tools* **9**(1), 25–36 (2004).
22. ISO/IEC JTC1/SC29/WG11, "1-D parallel test sequences for MPEG-FTV," MPEG document M15378 (2008).
23. ISO/IEC JTC1/SC29/WG11, "Philips response to new call for 3DV test material: Arrive book & mobile," MPEG document M16419 (2009).
24. ISO/IEC JTC1/SC29/WG11, "Description of exploration experiments in 3D video coding," MPEG document N11477 (2010).
25. G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Doc. VCEG-M33 (2001).

Cheon Lee received his BS degree in electronic engineering and avionics from Korea Aerospace University (KAU), Korea, in 2005 and an MS degree in information and communication engineering at the Gwangju Institute of Science and Technology (GIST), Korea, in 2007. He is currently working toward his PhD degree in the Information and Communications Department at GIST, Korea. His research interests include digital signal processing, video coding, 3D video coding, depth estimation, 3D television and realistic broadcasting.



Byeongho Choi received the B.S and M.S degrees in Electronic engineering from the University of Hanyang, Republic of Korea, in 1991 and 1993, respectively, and PhD degree in Department of Image Engineering from the University of Chungang, Republic of Korea, in 2010. From 1993 to 1997, he had worked for LG Electronics Co. Ltd as a junior researcher. In 1997, he joined Korea Electronics Technology Institute (KETI), where he was involved in the development of multi-view video, stereo vision and other video systems. He is currently a Managerial Researcher of SoC Research Center. His research interests include digital image processing, and its application, especially such as 3DTV, stereo vision system.

Yo-Sung Ho received both BS and MS degrees in electronics engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and a PhD degree in electrical and computer engineering from the University of California, Santa Barbara, in 1990. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 1983. From 1990 to 1993, he was with Philips Laboratories, Briarcliff Manor, New York, where he was involved in the development of the advanced digital high-definition television (AD-HDTV) system. In 1993, he rejoined the technical staff of ETRI and was involved in the development of the Korea direct broadcast satellite (DBS) digital television and high-definition television systems. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), where he is currently a professor in the Information and Communications Department. Since 2003, he has also been director of Realistic Broadcasting Research Center (RBRC) at GIST in Korea. His research interests include digital image and video coding, image analysis and image restoration, advanced coding techniques, digital video and audio broadcasting, 3D television, and realistic broadcasting.