MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

View Synthesis Prediction in the 3D Video Coding Extensions of AVC and HEVC

Zou, F.; Tian, D.; Vetro, A.; Sun, H.; Au, O.C.; Shimizu, S.

TR2014-055 March 2014

Abstract

Advanced multiview video systems are able to generate intermediate viewpoints of a 3D scene. To enable low complexity free view generation, texture and its associated depth are used as input data for each viewpoint. To improve the coding efficiency of such content, view synthesis prediction (VSP) is proposed to further reduce inter-view redundancy in addition to traditional disparity compensated prediction (DCP). This paper describes and analyzes rate-distortion optimized VSP designs, which were adopted in the 3D extensions of both AVC and HEVC. In particular, we propose a novel backward-VSP scheme using a derived disparity vector, as well as efficient signaling methods in the context of AVC and HEVC. Additionally, we put forward a novel depth-assisted motion vector prediction method to optimize the coding efficiency. A thorough analysis of coding performance is provided using different VSP schemes and configurations. Experimental results demonstrate average bit rate reductions of 2.5% and 1.2% in AVC and HEVC coding frameworks, respectively, with up to 23.1% bit rate reduction for dependent views.

IEEE Transactions on Circuits and Systems for Video Technology

View Synthesis Prediction in the 3D Video Coding Extensions of AVC and HEVC

Feng Zou, *Member, IEEE,* Dong Tian, *Member, IEEE,* Anthony Vetro, *Fellow, IEEE,* Huifang Sun, *Fellow, IEEE,* Oscar C. Au, *Fellow, IEEE,* and Shinya Shimizu

***Abstract*—Advanced multiview video systems are able to generate intermediate viewpoints of a 3D scene. To enable low complexity free view generation, texture and its associated depth are used as input data for each viewpoint. To improve the coding efficiency of such content, view synthesis prediction (VSP) is proposed to further reduce inter-view redundancy in addition to traditional disparity compensated prediction (DCP). This paper describes and analyzes rate-distortion optimized VSP designs, which were adopted in the 3D extensions of both AVC and HEVC. In particular, we propose a novel backward-VSP scheme using a derived disparity vector, as well as efficient signalling methods in the context of AVC and HEVC. Additionally, we put forward a novel depth-assisted motion vector prediction method to optimize the coding efficiency. A thorough analysis of coding performance is provided using different VSP schemes and configurations. Experimental results demonstrate average bit rate reductions of 2.5% and 1.2% in AVC and HEVC coding frameworks, respectively, with up to 23.1% bit rate reduction for dependent views.**

***Index Terms*—3D, video coding, depth, view synthesis prediction**

I. Introduction

THE past decade has witnessed an overwhelming proliferation of 3D video applications for both the movie industry and home entertainment due to rapid advancements in 3D multimedia technologies. For example, IMAX movie theaters [1] have gained a majority of 3D movie markets worldwide and offer a premium user experience. In this system, two separate images corresponding to the viewpoints of each eye are projected on to a special silver-coated screen at the same time. 3D glasses are used to separate the two images, and then the human brain blends them together to create an immersive 3D image sequence. In the consumer market, the manufacturing cost of 3D displays has been reduced due to improvements in LCD/LED manufacturing. As a result, 3D displays with stereoscopic capabilities have become available, and further advances will make multiview auto-stereoscopic displays commercially viable in the near future. As 3D content becomes more prevalent, the efficient compression, storage and transmission are pressing and challenging needs.

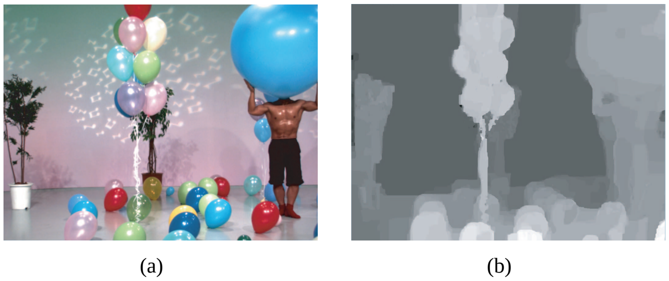

(a)
(b)

Fig. 1. Captured information from viewpoint 1 for sequence Balloons of size 1024x768: (a) texture image; (b) depth image.

To improve the 3D video coding efficiency, multi-view video coding (MVC) was developed as an important extension of AVC [2]. In MVC, a scene of interest is assumed to be captured through an array of densely placed time-synchronized cameras, without any captured depth. Instead of encoding each view separately, i.e., simulcast, MVC exploits the correlation between different views using inter-view prediction [3][4]. Although substantial rate savings can be achieved with MVC, the bit rate and complexity will increase linearly with the number of views. Also, there is no provision to enable generation of intermediate views, which is needed for free viewpoint applications or to generate the large number of views required for an auto-stereoscopic display.

To address these needs, a multi-view plus depth (MVD) data format, as shown in Fig. 1, is considered to facilitate intermediate view generation with low complexity. To represent this input data format efficiently, new standardization development efforts have been launched to assess and standardize a coding framework along with associated coding tools. One unique aspect of the evaluation process is that the quality of intermediate views is considered in the evaluation of coding efficiency. Extensions of both AVC and HEVC standards that support depth are now being developed.

To further improve the coding efficiency of 3D video coding system based on the MVD format, we propose a novel coding scheme that utilizes the depth information to efficiently code the texture data. The primary contribution of this paper is a novel view synthesis prediction (VSP) coding scheme that uses a derived disparity vector. This scheme has been integrated into both AVC and HEVC coding frameworks and realized using efficient signalling of the VSP coding modes. Another key contribution of this paper is a novel depth-assisted motion vector prediction technique to optimize the coding efficiency. An in-depth analysis of coding performance of

Feng Zou was taking internship at Mitsubishi Electric Research Laboratories while he was a Ph.D student at Hong Kong University of Science and Technology. Now he works at Qualcomm. (e-mail: fenzou@gmail.com)

Dong Tian and Anthony Vetro are with Mitsubishi Electric Research Laboratories, USA. (e-mail: {tian,avetro}@merl.com)

Oscar C. Au is with Hong Kong University of Science and Technology, Hong Kong SAR. (e-mail: eeau@ust.hk)

Shinya Shimizu is with NTT Media Intelligence Laboratories, Tokyo. (e-mail: shimizu.shinya@lab.ntt.co.jp)

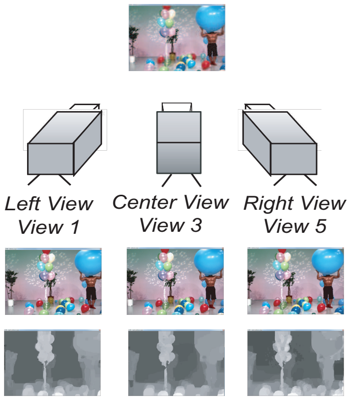


Fig. 2. An example of the three view rendering case, including left, center and right views.

different schemes and configurations is also provided.

The remainder of this paper is organized as follows. Section II provides a brief review of related work, including an introduction to principles and methods used to realize VSP. Section III introduces different VSP architectures and our proposed designs for both 3D-AVC and 3D-HEVC. A method for generating a derived disparity vector is discussed, and signalling aspects in both standardization frameworks are presented in this section. In Section IV, a depth-assisted motion vector predictor is put forward to further improve the 3D video coding efficiency. In Section V, extensive simulations are conducted to evaluate the performance of the proposed VSP schemes in each coding framework. Finally, concluding remarks are given in Section VI.

II. Related Work

A Joint Collaborative Team on 3D Video Coding (JCT-3V) was formally established in July 2012 by ITU-T and ISO/IEC to develop 3D video coding standards with more advanced compression capability and support for synthesis of additional perspective views; covering both AVC and HEVC based extensions. In this paper, the 3D extensions of AVC will be referred to as 3D-AVC, while the 3D extensions of HEVC will be referred to as 3D-HEVC. For this development, the MVD format was selected as the input data format representation, as shown in Fig. 1 since it is able to facilitate intermediate view generation using depth image-based rendering (DIBR) [5]. Typically, the MVD data format includes a selection of texture videos and their corresponding depth captured in a time synchronized manner from different viewpoints as shown in Fig. 2. A receiver can choose appropriate reconstructed views to interpolate the intermediate views of user's interest according to the geometric information conveyed in the reconstructed depth components. From the coding perspective, the inclusion of depth in addition to texture in the input data poses a new challenge as more data needs to be coded. Interestingly, the depth and the texture components have a mutually beneficial relationship in that the depth can be used to provide a good alternative prediction of the texture while the reconstructed texture can serve as a good structural description of the depth. Motivated by these two features, two categories of research have been conducted to improve the overall coding efficiency of MVD systems.

In the first category of research, depth data is coded with the help of reconstructed texture data. While the depth data has considerably different signal characteristics than the texture data, it does exhibit some structural similarity to the corresponding texture. For instance, an edge in the depth component usually corresponds to an edge in the texture component. However, a minor distortion of the depth value can result in considerable distortion in the synthesized texture using DIBR techniques. Such errors can be especially serious at edge locations in the depth component [6]. To mitigate this problem, advanced tools for coding depth to better preserve edges in the depth component have been proposed, e.g., intra prediction using wedgelet or contour partitions [7][8][9] [10]. Furthermore, to exploit the similarity in motion characteristics between the texture and depth, it has been proposed to inherit motion from the corresponding texture component [9][11], thus saving the overhead bits to encode motion for the depth.

In the second category of research, depth data is utilized to provide an alternative disparity-compensated predictor in addition to the traditional motion-compensated predictors. Specifically, with the MVD data format, it becomes possible to support the generation of intermediate views at the receiver using DIBR, whereby intermediate views are generated by using depth. While DIBR is typically used as a post-processing step to generate intermediate synthesized views for output and display, it was proposed in [12] to utilize this technique to provide an alternative non-translational pixel-based disparity-compensated predictor for each block in the coding loop. This in-loop technique is commonly referred to as view synthesis prediction (VSP).

As such, 3D video coding with depth supports three possible predictors: traditional block-based MCP, block-based translational DCP and the pixel-based non-translational VSP. To realize VSP, it was proposed that a synthesized picture be added to the reference picture list before encoding the current view [12][13][14]. [15] further proposed a rate-distortion optimized VSP by incorporating a block-based depth correction vector. A scalable enhancement view predictor is also proposed in [16], where the base views and the residue of enhancement views are encoded by a conventional video coding process. In [17], a general VSP scheme has been developed that extends the warping source from one view to two views, and applies VSP to both texture and depth components.

While prior work on this topic has shown promising results, the level of performance and validation has not been sufficient to be incorporated into any of the previously developed video coding standards. Building on this earlier work, the following section describes the details of our proposed designs for VSP, which are able to realize notable and consistent gains in coding efficiency. Additionally, the designs have become practical and have been rigorously evaluated. As a result, the VSP concept has been adopted into 3D extensions of both the AVC and HEVC coding standards.

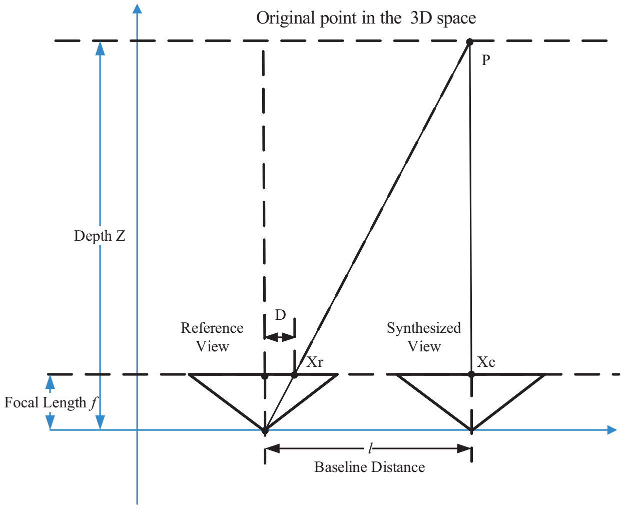


Fig. 3. Geometric relationship in VSP using depth image-based rendering (DIBR)

III. Proposed VSP Designs

In this section, two VSP designs are considered, each one being suited to a particular coding architecture or constraint. The first design uses depth from the reference view to perform a forward warping operation, and is hence referred to as forward VSP (FVSP), while the second design uses depth of the current view to perform a pixel-based warping and is referred to as backward VSP (BVSP).

Due to the inherent dependence on depth, the selection of the FVSP or BVSP design highly depends on the coding order of texture and depth components. When the depth component is coded prior to its corresponding texture component, i.e., depth-first coding order, BVSP can be directly implemented. In contrast, when the depth component is coded after its corresponding texture component, i.e., texture-first coding order, only FVSP can be directly implemented.

In this section, we will briefly review and analyze these two designs. Then, we propose a novel scheme to realize the more implementation-friendly BVSP design with texture-first coding [18], which is adopted in the 3D-HEVC standard.

A. Comparison of Forward and Backward VSP

In this subsection, we discuss the basic concept of VSP in a 3D video coding system with MVD as input. Two assumptions are made: first, the cameras are placed in a 1D array and are rectified; second, the object surface is a Lambertian surface, that is, a point in the surface has identical intensity values from different viewpoints. Under these two assumptions, VSP synthesizes a virtual view from a reference view by applying 3D warping using depth information, and the synthesized view is used as a predictor for the current view.

In 3D geometry, a pixel S_r at a location X_r in the reference picture corresponds to an object surface point P as shown in Fig. 3. Here we use the subscripts r and c to indicate quantities from the reference view and the current view respectively. The depth value d_r associated with P has the following relationship with the actual distance value Z,

$$\frac{1}{Z} = \frac{d_r}{255} \cdot \left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + \frac{1}{Z_{far}} \tag{1}$$

where Z_{near} and Z_{far} are the smallest and largest actual distance among all surface points captured by the camera. Let P correspond to a pixel S_c at a location X_c with depth d_c in the current frame (to be synthesized). Using triangular similarity, the disparity value (horizontal displacement) D between X_r and X_c should be

$$D = f \cdot l / Z \tag{2}$$

where f is the common camera focal length of the reference camera and the current camera and l is the baseline distance between them. Therefore, in VSP, the surface point P in the 3D scene is rendered at position X_c in the synthesized view with

$$X_c = X_r - D \tag{3}$$

And the pixel value S_r at X_r is copied to S_c at X_c in the synthesized view.

$$S_c(X_c) = S_r(X_r) \tag{4}$$

In practice, there are two ways to implement VSP, which depends on whether d_r is used (forward VSP) or d_c is used (backward VSP). Further details are described below.

With forward VSP, the d_r values are used to compute the disparity of each pixel S_r and to warp each pixel S_r from the reference view to the current synthesized view S_c using (2)-(4). After all reference pixels are warped, there are typically some vacant pixels, or holes, in the current view without any assigned value, mainly due to object occlusion. Typically, inpainting methods are used to fill the holes. As only information of the reference view texture and depth are used, the synthesized frame generated by FVSP can be stored in the reference frame list in a hybrid video framework before encoding the current view.

There are two main drawbacks of FVSP. Firstly, the hole filling process requires hole pixel identification and value assignment in a sequential order, because the processing of one pixel depends on its preceding pixel. And thus the hole filling process requires additional memory for hole indication, conditional checking for pixel availability, and irregular memory access. All these can lead to irregular dataflow, broken pipeline, higher memory requirements, and higher power consumption. Parallel processing is not feasible due to the sequential processing order of these operations. Secondly, FVSP generates the entire synthesized frame non-discriminatively. While this is reasonable for the encoder as the encoder needs to try all different prediction modes during the mode decision, it is a waste of decoder computation to unnecessarily generate the synthesized pixels for those blocks that do not choose VSP.

With backward VSP, as presented in [15], it is assumed that the depth of the current view d_c is available and is used to compute the disparity D of each pixel S_c at X_c as shown in Fig. 4. On finding the corresponding reference pixel S_r at location $X_r = X_c + D$ in the reference view, it simply copies S_r to S_c. As such, BVSP does not inherit the two drawbacks

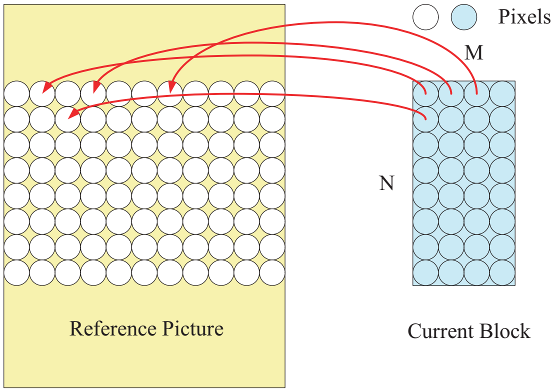


Fig. 4. An example of block based backward view synthesis prediction (BVSP). Each pixel in the current block of size $N \times M$ has a corresponding reference pixel represented by disparity vector. Typically, the reference picture denotes the base view while the current block is from the dependent view.

of FVSP in that each pixel can always find a reference pixel, so there are no holes in the synthesized frame, and since there is no hole filling, BVSP has significantly lower complexity than FVSP. The BVSP design is amenable to parallel block-based processing and can avoid generating unnecessary synthesis blocks at the decoder.

It is evident that BVSP requires a depth-first coding order, e.g., $\mathrm{T_0D_0D_1D_2T_1T_2}$, where T_i and D_i represent the texture and depth respectively from the i^{th} view in the view coding order, since the depth of the current view is needed when coding the texture of the current view. As 3D-AVC supports both texture-first and depth-first coding orders, both FVSP and BVSP designs were studied and BVSP was finally being chosen considering that it has a similar coding performance as FVSP and facilitates a more practical design [21][22].

In contrast to 3D-AVC, 3D-HEVC currently assumes a texture-first coding order, e.g. $\mathrm{T_0D_0T_1D_1T_2D_2}$, which prohibits BSVP from being directly applied. In order to support VSP in the 3D-HEVC framework, we initially proposed an FVSP design [23][24]; however, this was not adopted due to the large decoder complexity increase. To reduce the additional decoder complexity, we proposed a novel BVSP design using derived disparity [18] for 3D-HEVC. This scheme showed comparable coding gains relative to the FVSP design with much lower complexity, and was adopted into the 3D-HEVC working draft in January 2013.

B. BVSP with derived disparity

In this subsection, we propose a novel BVSP scheme with derived disparity that estimates a current view depth block d_c using the spatial correlation in d_c and the available reference view depth d_r, which is the major challenge of BVSP.

When the texture-first coding order prohibits the generation of a BVSP reference, we apply two approximations to estimate the depth of the current view d_c. The first approximation is that we use the disparity vector of the neighboring block [19] to approximate the disparity vector of the current block such that a reference block can be localized with the disparity vector pointing to the reference view, see Step 1 of Fig. 5. This approximation is reasonable since the motion information[1] of neighboring blocks bears close resemblance with that of the current block, and it provides a good motion predictor in existing video coding standards. The second approximation is that we use the corresponding depth block in the reference view as the estimated depth block for the current view, see Step 2 of Fig. 5. The second approximation is valid when different views are capturing the same scene and the cameras are not too far away from each other. Given the estimated depth block, BVSP can be implemented by fetching reference pixels in the reference view as shown in Step 3 of Fig. 5. Specifically, the per pixel depth value is converted to per pixel disparity vector according to the geometric information between two views using (1) and (2). With the pixel based disparity vector, a reference pixel can be fetched from the reference texture picture. And all the fetched reference pixels form a VSP predictor for the current block. To limit the complexity from per-pixel disparity compensation, a block based BVSP is preferred, where a single disparity is converted from a representative depth value for the whole block. The block size used for BVSP has been refined from 4x4 [18] to 4x8/8x4 [25].

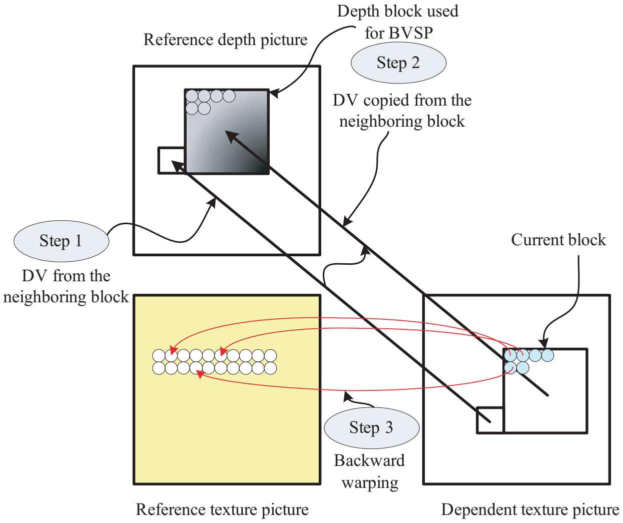


Fig. 5. Proposed BVSP using the disparity vector derived from the neighboring blocks.

It is noted that we originally proposed the BVSP using neighboring blocks in the texture first coding order in the context of 3D-HEVC. At a later JCT3V meeting, the same approach to support BVSP with texture first coding order was adopted into 3D-AVC [20].

C. Efficient signalling of VSP modes in 3D-HEVC

In this subsection, to efficiently represent the proposed VSP mode for each block (in either FVSP or BVSP), a VSP merge candidate is proposed to be included in the merge candidate list for both Skip and Merge modes in 3D-HEVC.

Recall that HEVC specifies Skip and Merge modes to inherit the motion information from spatially or temporally

[1] Here the motion information includes both the temporal motion information and the inter-view disparity information.

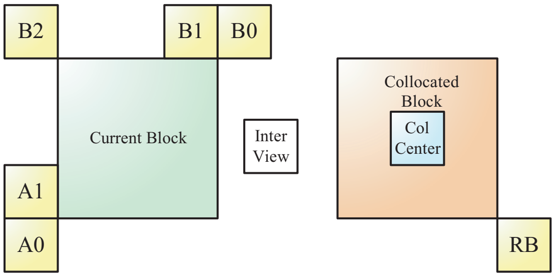


Fig. 6. Merge Candidate List for 3D-HEVC. The difference between 3D-HEVC and HEVC lies in the additional inter-view motion candidate derived from neighboring view in 3D-HEVC as of HTM5.1.

neighboring blocks to form a motion merged region. In particular, the motion information from spatial and temporal neighboring blocks, as shown in Fig. 6, composes a merge candidate list. In HEVC, each prediction unit (PU), if using the Skip or Merge mode, transmits a merge index to indicate the selection decision among the available merge candidates, from which the motion information is inherited. For traditional Inter mode (non-skip and non-merge mode), HEVC allows the encoder to choose a motion vector predictor among several motion vector predictor (MVP) candidates (similar definition as the merge candidates), and then motion vector difference, reference frame index, reference list and the predictor index are coded as motion information in the bitstream.

Similar with HEVC, 3D-HEVC has three types of inter modes for inter frame coding, namely Skip mode, Merge mode and Inter mode. The key difference between 3D-HEVC and HEVC lies in the addition of inter-view motion information prediction [26][19] included in both the merge list and the motion vector predictor list in 3D-HEVC. And the merge list consists of up to six merge candidates $M = \{m_k | k = 0, 1, ..., 5\}$ including spatial, temporal neighboring motion vector predictors and the inter-view motion prediction.

To efficiently represent VSP, a VSP merge candidate is proposed to be included in the merge candidate list with (0,0) motion pointing to the synthetic reference block generated by FVSP or BVSP. In other words, if the current block chooses VSP, the synthetic reference block is directly used as the compensated block. Note that the maximum number of allowable merge candidates is proposed to be unchanged as six. In other words, the first six available candidate (in a predefined order) are used to constitute the merge candidate list. Though, a single VSP merge candidate may be inserted during the candidate construction process, it was adopted later to allow more VSP merge candidates to be inherited from neighboring blocks coded with VSP mode [27]. The inherited VSP merge candidates would use the disparity vector carried from its neighboring block to fetch the depth block and then conduct VSP prediction.

At the encoder, the merge index k is decided based on the rate-distortion cost for each candidate

$$m_{k^*} = arg\ \min_{m_k} \|X_{org} - X_{pred}(m_k)\|^2 + \lambda \times R(m_k) \quad (5)$$

where X_{org} and $X_{pred}(m_k)$ are the original signal and compensated predictor using the motion predictor candidate m_k. λ is a predefined Lagrangian multiplier depending on Quantization Parameter QP. R stands for the number of bits to code the block when using the merge candidate k.

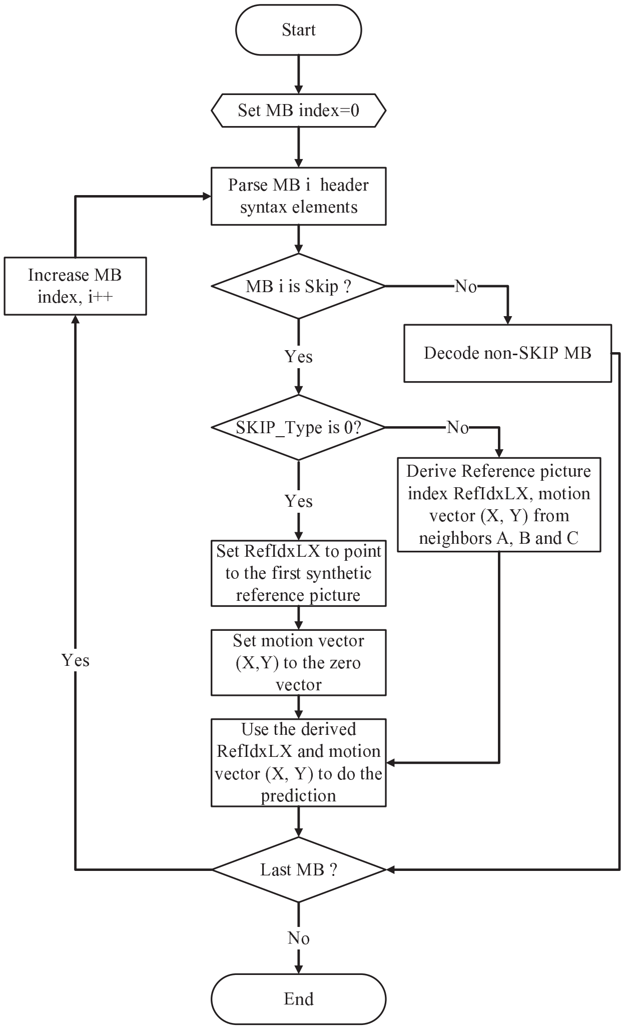

Start
Set MB index=0
Parse MB i header syntax elements
MB i is Skip ?
No
Decode non-SKIP MB
Yes
SKIP_Type is 0?
No
Derive Reference picture index RefIdxLX, motion vector (X, Y) from neighbors A, B and C
Yes
Set RefIdxLX to point to the first synthetic reference picture
Set motion vector (X,Y) to the zero vector
Use the derived RefIdxLX and motion vector (X, Y) to do the prediction
Last MB ?
Yes
Increase MB index, i++
No
End

Fig. 7. Proposed VSP SKIP work flow in the context of 3D-AVC. Similar work flows applies for VSP Direct mode.

D. Efficient signalling of VSP modes in 3D-AVC

In this subsection, we describe the efficient signalling of Skip and Direct modes with respect to VSP references in 3D-AVC [28], which follows a similar concept of the VSP merge candidate proposed in 3D-HEVC. Recall that in AVC, there is a significant portion of macroblocks chosen as Skip modes, where there is neither motion vector difference (MVD) nor residue coefficients in P frames, and Direct modes, where there is no MVD but residue coefficients in B frames. It is also observed that the usage of Skip and Direct modes increases with lower bit rate. Therefore, to more efficiently represent the VSP Skip and Direct modes, concepts are inherited from AVC Skip and Direct modes. The main difference is that the predictor is obtained from the synthesized block and the motion vector corresponding to a VSP reference picture is assumed to be (0,0) as the geometric mapping results are assumed to be well aligned with the current block. To signal the VSP Skip mode, the existing SKIP_Flag is used to differentiate the Skip mode and non-Skip mode. If SKIP_Flag is 1, an additional syntax SKIP_Type is employed to differentiate VSP Skip and non-VSP Skip modes. A similar signalling is applied to Direct modes in B frames using an additional syntax element, DIRECT_Type. A typical work flow is shown in Fig. 7. Note that the proposed signaling method is applicable for both FVSP and BVSP. In BVSP, the synthetic reference block is treated as if it was from a synthetic reference picture, although there is no need to create a physical reference frame buffer.

Compared with 3D-AVC, our proposed scheme adds one additional macroblock (MB) level syntax (SKIP_Type) in the bitstream. At the encoder, the SKIP_Type decision is made to minimize the Lagrangian cost J as follows:

$$\begin{aligned} J = \min_{SKIP\_Type=\{0,1\}} & [SSE(SKIP\_Type) \\ & + \lambda \times R(SKIP\_Type)] \end{aligned} \tag{6}$$

where λ is the predefined Lagrangian multiplier in AVC, SSE represents the reconstruction error, and R denotes the bits to encode current MB including SKIP_Type. A similar cost function is used for VSP Direct modes.

IV. Proposed Depth-assisted motion vector predictor for inter-view motion prediction

In this section, we revisit traditional block-based motion estimation (ME) in a rate distortion (RD) optimized video coding framework and point out that the motion vector decision during ME depends not only on matching accuracy (the distortion), but also on the rate to code the motion information. To achieve the smallest possible RD cost, we propose a depth-assisted motion vector predictor (MVP) and argue that when the motion vector of the current block is chosen as the depth-assisted MVP, the lower bound of the RD cost is achieved.

First, recall that in traditional block-based hybrid video coding, ME is used to find a block Y of size $N \times M$ in a reference frame that is similar to the current block X of size $N \times M$. The relative displacement between X and Y is represented by a two dimensional (vertical and horizontal) displacement vector called motion vector $MV = (v_x, v_y)$. Often, the optimal MV, or MV_{opt}, in RD-optimized video coding is the $MV \in S$ that minimizes the following RD cost

$$C = SAD(MV) + \lambda R(MV - MVP) \tag{7}$$

where S is the motion vector candidate set, SAD is the sum of absolute difference between the input block and the predictor pointed by MV, MVP is the motion vector predictor, R is the rate to encode the motion vector difference and λ is a positive Lagrangian multiplier. Here, for simple explanation, we assume that the other signaling costs, such as reference frame index, reference list and motion vector predictor index, are the same. And thus those signaling are not included in (7). Also note that C can be considered as a function of both MV and MVP. We note that both AVC and HEVC use block-based translational motion-compensated prediction to reduce inter frame redundancy and the MV is always a block-based temporal motion vector. But 3D-HEVC supports three kinds of predictors for the current block: block-based translational motion compensated predictor, block-based translational disparity-compensated predictor, and pixel-based non-translational disparity-compensated predictor (VSP). The search for MV_{opt} mentioned above is applied to both kinds of "block-based" predictors such that the MV can be a block-based temporal motion vector or a block-based disparity vector, and the S contains all possible block-based temporal motion vectors and all possible block-based disparity vectors.

To reduce the neighboring block motion redundancy, there are typically a number of MVP candidates, MVP_k, for $k = 1, 2, 3..., K$, generated from neighboring inter coded blocks. Usually $R(.)$ is a non-negative convex function with the global minimum achieved at zero. Therefore for each MVP_k, the minimum of the second term λR in (7) is achieved when $MV = MVP_k$. In other words,

$$\text{arg} \min_{MV \in S} [\lambda \times R(MV - MVP_k)] = MVP_k \tag{8}$$

Let MV_{SAD} be the motion vector that achieves the minimum of the first term SAD in (7) such that

$$MV_{SAD} = \text{arg} \min_{MV \in S} SAD(MV) \tag{9}$$

Note that MV_{SAD} does not depend on the MVP_k, and $C \geq SAD(MV_{SAD})$.

Suppose we have the freedom to choose the MVP_k definition, and we choose $MVP_{k'}$=MV_{SAD} for one of the k, with $k = k'$. Then the $MV = MVP_{k'}$ can simultaneously minimize both the first term SAD and the second term λR of (7). In fact, the second term λR will become $\lambda R(0)$ which is the lower bound of the term. In other words, the choice of $MVP_{k'}$=MV_{SAD} and $MV = MVP_{k'}$ achieves the smallest possible value for C among all the possible MVP_k.

Note that MV in (7) is a block-based motion vector. It is restrictive in the sense that all pixels at any location (i, j) within the current block X must have the same motion vector MV. If different pixels can have different motion vectors, the minimum achievable C can perhaps be smaller. Here we will take advantage of the third kind of predictor, pixel-based non-translational disparity-compensated predictor (VSP), in an attempt to achieve lower C.

Let U be the collection of all the pixel-level motion vectors $MV_{i,j}$ in the current block.

$$U = \{MV_{0,0} \ , \cdots, MV_{N,M}\} = \{MV_{i,j}\}_{i=1,j=1}^{N, \ \ M} \tag{10}$$

Then finding MV_{opt} that minimizes (7) is equivalent to finding U_{opt}

$$U_{opt} = \text{arg} \min_{U \in S_G} [SAD'(U) + \lambda \times R'(U - U_P)] \tag{11}$$

where U_P is the collection of pixelwise predictors for U with all pixelwise predictors being equal, R' is the pixelwise rate, SAD' is the pixelwise SAD and the feasible set S_G contains only those candidates U with all pixelwise MV being equal.

Now, let us consider a more general case in which each pixel (i, j) is allowed to have its own motion vector $MV_{i,j}$ (i.e. disparity vector in the case of VSP, which will be discussed in the following). Then we define U_{opt} as that in (11) except that we define S_G to be all candidate collections of disparity vectors and U_P to be the set of disparity predictors. Note that the components of S_G and U_P do not need to be equal. Similar to the past, we assume $R'(U - U_P)$ to be a non-negative function which is convex with respect to each component, with minimum achieved when $(U - U_P) = \mathbf{0}$.

Let U_{SAD} be the collection of pixelwise disparity vectors that achieves minimum SAD'. In other words,

$$U_{SAD} = \arg \min_{U \in S_G} SAD'(U) \tag{12}$$

Suppose we have the freedom to choose the U_P definition, and we choose $U_P = U_{SAD}$. Then the $U = U_P$ can simultaneously minimize both the first term SAD' and the second term $\lambda R'$ of (11) over U. In fact, the second term $\lambda R'$ will become $\lambda R'(\mathbf{0})$ which is the lower bound of $\lambda R'(.)$. In other words, the choice of $U_P = U_{SAD}$ and $U = U_P$ achieves the smallest possible value of (11) among all the possible U_P.

In the following, we argue that when we use the depth information to generate U_P and choose $U = U_P$ in VSP, the global minimum C is indeed achievable under certain assumptions.

In particular, we make three assumptions: the depth is of high accuracy; the object surface is a Lambertian surface; and there is no occlusion effect for the current block. Under these assumptions, when we convert each pixel's depth $Dep_{i,j}$ to its disparity vector $Dis_{i,j} = (v_x^{i,j}, v_y^{i,j})$ in VSP, a reference pixel is located from the inter-view reference picture as a perfect match with the current pixel in terms of **zero** absolute difference. Thus, when we choose $U_P = \{Dis_{i,j}\}_{i=1,j=1}^{N,\ M}$, U_P minimizes the SAD' as follows

$$\min_{U \in S_G} SAD'(U) = SAD'(U_P) = 0 \tag{13}$$

And (13) is equivalent to (14) as

$$U_P = U_{SAD} \tag{14}$$

Thus the choice of $U_P = \{Dis_{i,j}\}_{i=1,j=1}^{N,\ M}$, and $U = U_P$ achieves the lower bound of (11) with a corresponding cost $0 + \lambda R'(\mathbf{0})$. Since SAD' and $\lambda R'$ are both non-negative, $0 + \lambda R'(\mathbf{0})$ is a global minimum. In other words, when we use the per pixel disparity as the per pixel disparity vector predictor, and the VSP predictor as the pixel-based non-translational disparity compensated predictor, the global minimum ME cost is achieved among all the three kinds of predictors mentioned above.

Motivated by the superior RD property of VSP, we propose to use $\{Dis_{i,j}\}_{i=1,j=1}^{N,\ M}$ to generate a representative block-based disparity vector predictor MVP, for those blocks using block-based translational disparity compensated prediction, such that $U_P = \{\{MVP_{i,j}\}_{i=1,j=1}^{N,\ M} | MVP_{i,j} = MVP\}$ is a good approximation of $\{Dis_{i,j}\}_{i=1,j=1}^{N,\ M}$. And the generation process is formulated as

$$MVP = f(\{Dis_{i,j}\}_{i=1,j=1}^{N,\ M}) \tag{15}$$

where the function $f(\cdot)$ can be any function generating one MVP, i.e., $max(\cdot)$, $mean(\cdot)$, $min(\cdot)$ and etc. Usually, $max(\cdot)$ is used because it can capture the foreground object disparity very well and the background matching error tends to be small when assuming the background is smooth, which is usually the case. As such, $max(\cdot)$ is selected as a function to generate a representative disparity for each block.

Compared with traditional MVP, the depth-assisted MVP is not decided according to the RD criteria, but rather derived directly from the geometric information conveyed in the depth. In case of the VSP mode, the depth-assisted MVP can achieve the lower bound of the RD cost. What is more, the depth-assisted MVP does not require additional overhead, as it can be derived from the coded depth block. Therefore, the depth-assisted MVP (block or pixel based) is free to be utilized at both the encoder and decoder. And the depth-assisted MVP process is used in both Merge mode and traditional inter mode (AMVP) during motion vector prediction.

It is worth mentioning that a similar approach called depth-oriented neighboring block disparity vector (DoNBDV) [29] is developed independently from VSP to improve the motion vector predictor accuracy. The major difference between these two techniques is that the original DoNBDV proposed in [29] provides a refined MVP for a conventional interview block only, while our depth-assisted MVP targets to harmonize the interactions between the VSP mode (a new interview mode) and the conventional interview mode. In other words, with depth-assisted MVP, we propose to derive a proper MVP from a neighboring VSP coded block for the current block when it is coded either in VSP mode or conventional interview mode. Without the depth-assisted MVP we have to disable the MVP from neighboring VSP blocks, and it would degrade the performance of both traditional interview block and VSP block due to poor MVP prediction. So, the final adopted DoNBDV in 3D-HEVC is actually a combination of depth-assisted MVP [18] and the original DoNBDV in [29].

V. Experimental Results

This section describes the experimental results using the proposed VSP schemes in both 3D-AVC and 3D-HEVC test models. Simulations were performed under the common test conditions defined by JCT-3V in [30] where the test set includes 7 video sequences of size 1024×768 and 1920×1088 with the MVD data format. Variations of VSP are discussed and compared objectively followed by a VSP usage analysis for both 3D-AVC and 3D-HEVC. Finally, a complexity comparison is conducted to evaluate different VSP schemes on 3D-AVC and 3D-HEVC platforms.

A. VSP and its efficient signalling in 3D-AVC

Generally speaking, 3D-AVC inherits the basic hybrid predictive video coding structure of AVC. In the development of 3D-AVC, a hierarchical B coding structure is used to exploit the temporal inter-frame redundancy while IPP coding structure is used to exploit the inter-view redundancy as shown in Fig. 8. Specifically, at each time instance, there are three views, including a center view, a left view and a right view. Since the center view is near to both left and right views, it is coded first (usually called base view), as it tends to provide a good reference for both the other two views (usually called dependent views). To independently decode the base view, the base view can only refer to the previously coded base views as

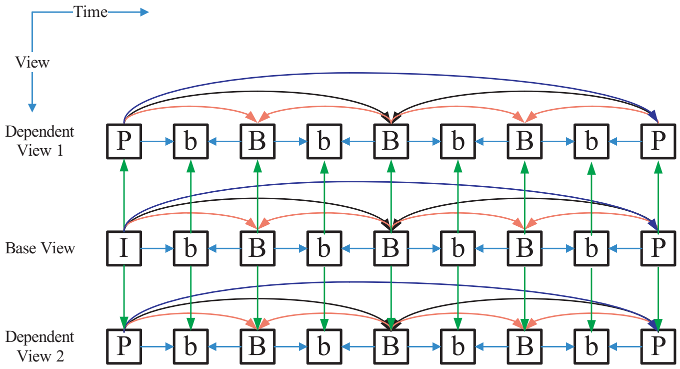


Fig. 8. An example of the three view coding structure

TABLE I
RD PERFORMANCE WITHOUT VSP SKIP/DIRECT AGAINST ANCHOR

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	-7.3%	-7.3%	-1.8%	-1.5%	-2.1%	117.3%	114.6%
Kendo	0.0%	-7.8%	-8.4%	-2.4%	-1.7%	-2.4%	108.8%	112.5%
Newspaper_C	0.0%	-2.7%	-2.3%	-0.6%	-0.5%	-0.7%	107.0%	107.6%
GT_Fly	0.0%	-23.1%	-21.5%	-5.3%	-5.0%	-4.3%	106.0%	112.1%
Poznan_Hall2	0.0%	-3.1%	-4.0%	-1.1%	-1.0%	-1.2%	113.0%	105.7%
Poznan_Stree	0.0%	-13.2%	-11.2%	-3.3%	-3.0%	-2.7%	107.8%	110.9%
Undo_Dancer	0.0%	-11.5%	-9.9%	-2.9%	-2.6%	-2.6%	110.4%	103.9%
1024x768	0.0%	-5.9%	-6.0%	-1.6%	-1.2%	-1.7%	111.1%	111.6%
1920x1088	0.0%	-12.7%	-11.6%	-3.2%	-2.9%	-2.7%	109.3%	108.1%
average	0.0%	-9.8%	-9.2%	-2.5%	-2.2%	-2.3%	110.1%	109.6%

reference. The dependent view can refer to both the previously coded base view and its previously coded temporal frames as reference.

Recall that each view point consists of a texture component and a depth component. In the standardization work of 3D-AVC, $T_0D_0D_1D_2T_1T_2$ coding order is used, where T_i and D_i are the texture and depth components respectively from the i^{th} view. For example, for the *Balloons* test sequence, the view coding order 0->1->2 corresponds to view3->view1->view5 (center->left->right). With this coding order, the texture from the base view and the depth from all views are available prior to the coding of the dependent view textures T_1 and T_2 and BVSP is applied [21][22] in ATM6.0 to both T_1 and T_2 within the same time instance.

To efficiently represent the VSP mode, we apply the proposed VSP design for Skip and Direct modes at the macroblock (MB) level; as described earlier, this is similar to the traditional Skip and Direct modes except that the predictors are generated from the BVSP process. The proposed VSP scheme has been accepted into the 3D-AVC reference software ATM6.0 [31]. To test the proposed VSP design, we turn VSP off in ATM6.0 and use it as "anchor". We repeat the experiment with VSP on in ATM6.0 and call it "VSP". We test the proposed VSP with four quantization parameters (QP) {26, 31, 36, 41} for the base view, where the same QP is used for texture and depth.

The Bjontegaard delta bitrate (BD-BR) [32] is used as the objective evaluation for the coding performance, with a negative value indicating the relative bitrate reduction compared with the anchor. Simulation results are shown in Table I, where "video 0" indicates the coding performance of the base view; "video 1" and "video2" indicate the coding performance of two dependent views respectively; "video PSNR vs. video bitrate" indicates the coding performance of three coded textures; "video PSNR vs. total bitrate" indicates the coding performance of coded textures over total bitrate (depth+texture); and "synthesis PSNR vs. total bitrate" indicates the coding performance of synthesized views over total bitrate. For *Balloons*, the view coding order 0->1->2 corresponds to view3->view1->view5, and the six generated intermediate views for *Balloons* are views 1.5, 2, 2.5, 3.5, 4 and 4.5.

In Table I, 2.5% bitrate reduction for coded views and 2.3% bitrate reduction for synthesized views are achieved when turning on the proposed VSP Skip and Direct modes. And there is up to 23.1% bitrate reduction for dependent views. Table I also demonstrates that the proposed VSP scheme incurs 10% encoder/decoder complexity increase compared with the anchor.

To further improve the signaling of the VSP SKIP_Type flag, a neighboring context-based skip flag position method is proposed in [33]. The basic idea is that if both the top and left neighboring MBs use VSP Skip, SKIP_Type is firstly coded to indicate whether it is VSP Skip, and then is followed by Skip flag to indicate whether it is Skip coded. Doing so would reduce the overhead bits to indicate VSP Skip from two bins to one if neighboring MBs use VSP Skip and the current MB does too. It is reported that additional 0.82% and 1.06% bitrate reduction is achieved for coded views and synthesized views respectively.

From experiments, it is found that a large percentage of MBs tend to choose VSP modes in anchor frames (the dependent views whose base view is coded as intra frame in a group of pictures (GOP)). Motivated by this phenomena, we also proposed a flexible frame level VSP scheme by enabling or disabling the VSP mode according to the frame type. When enabling VSP on the anchor frame only, there is on average 0.74%, and 0.67% bitrate reduction achieved for coded views and synthesized views respectively. Although the coding gain of VSP is reduced when it is applied only on anchor frames, the encoding/decoding complexity is reduced and becomes comparable with the anchor in terms of run time.

B. FVSP for 3D-HEVC

In general, 3D-HEVC inherits the basic hybrid predictive video coding structure of HEVC, but it allows inter-view and inter-component (depth-texture) prediction. In the development of 3D-HEVC, the same coding structure as that of 3D-AVC is used as shown in Fig. 8. However, in 3D-HEVC, a different coding order is currently assumed, i.e., $T_0D_0T_1D_1T_2D_2$, where the texture component is coded prior to its depth component for each view at each time instance. With this coding order, T_0 and D_0 are available when coding T_1/D_1 or T_2/D_2. In this case, we propose to apply FVSP by warping the reconstructed base view $\widetilde{T_0}$ to the dependent view T_1 or T_2 using reconstructed base view depth $\widetilde{D_0}$. And the proposed FVSP scheme is summarized as follows:

- Code base view texture T_0 and depth D_0

TABLE II
RD PERFORMANCE OF REFERENCE FRAME FVSP BASED ON HTM5.1

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	0.3%	2.1%	0.6%	0.6%	0.5%	111.4%	189.8%
Kendo	0.0%	0.5%	1.2%	0.4%	0.5%	0.4%	115.1%	197.8%
Newspaper_C	0.0%	1.4%	1.6%	0.6%	0.6%	0.8%	118.7%	215.6%
GT_Fly	0.0%	-6.0%	-6.4%	-1.6%	-1.5%	-1.5%	106.0%	190.1%
Poznan_Hall2	0.0%	-0.4%	0.3%	0.1%	0.2%	0.0%	98.1%	174.9%
Poznan_Stree	0.0%	-2.1%	-2.0%	-0.6%	-0.5%	-0.4%	103.8%	200.4%
Undo_Dancer	0.0%	-13.2%	-12.3%	-3.5%	-3.5%	-3.1%	111.2%	201.5%
1024x768	0.0%	0.7%	1.6%	0.6%	0.6%	0.5%	115.0%	201.1%
1920x1088	0.0%	-5.4%	-5.1%	-1.4%	-1.3%	-1.2%	104.8%	191.7%
average	0.0%	-2.8%	-2.2%	-0.6%	-0.5%	-0.5%	109.2%	195.7%

TABLE III
RD PERFORMANCE OF FVSP BASED ON HTM5.1

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	0.0%	2.0%	0.6%	0.5%	0.4%	102.6%	183.4%
Kendo	0.0%	0.3%	0.9%	0.4%	0.4%	0.2%	102.0%	183.4%
Newspaper_C	0.0%	1.4%	1.5%	0.6%	0.6%	0.6%	105.7%	201.4%
GT_Fly	0.0%	-7.0%	-7.8%	-1.8%	-1.8%	-2.2%	101.8%	192.8%
Poznan_Hall2	0.0%	-0.3%	0.0%	0.1%	0.2%	-0.3%	97.8%	181.9%
Poznan_Stree	0.0%	-3.0%	-2.4%	-0.7%	-0.7%	-0.6%	103.7%	208.3%
Undo_Dancer	0.0%	-13.7%	-12.8%	-3.6%	-3.6%	-3.2%	98.2%	185.3%
1024x768	0.0%	0.6%	1.5%	0.5%	0.5%	0.4%	103.4%	189.4%
1920x1088	0.0%	-6.0%	-5.7%	-1.5%	-1.5%	-1.5%	100.4%	192.0%
average	0.0%	-3.2%	-2.6%	-0.6%	-0.6%	-0.7%	101.7%	190.9%

- Warp the reconstructed base view $\widetilde{T_0}$ to the dependent target views using the reconstructed depth $\widetilde{D_0}$ of the base view
- Set the warped view as a reference frame in DPB when coding the dependent views

Similar procedures are applied to the depth component in our proposed FVSP. The proposed scheme is implemented on top of the 3D-HEVC reference software HTM5.1[34] and tested using four QPs {25, 30, 35, 40} for the base view texture. Simulation results are shown in Table II, which indicates that the proposed FVSP scheme with an additional synthesized reference frame provides 2.8% bitrate reduction on average for the dependent view 1 and 2.2% bitrate reduction on average for the dependent view 2, and 0.5% bitrate reduction for synthesized views is achieved. As a matter of fact, the performance is limited since the VSP mode is only initiated from the traditional inter mode (where reference frame index, prediction direction, and MVD are transmitted), rather than Skip or Merge modes. When the VSP merge candidate is inserted in the merge candidate list for Skip and Merge modes, 3.2% and 2.6% bitrate reduction are achieved for dependent view 1 and 2. For synthesized views, 0.7% bitrate reduction is obtained as shown in Table III.

Note that the results mentioned above are obtained when the proposed FVSP scheme is applied to both texture and depth. When it is applied to texture only, the simulation results are shown in Table IV. Comparing Table III and Table IV, we conclude that applying VSP on texture benefits the coding performance of both coded and synthesized views, and applying VSP on depth would further improve the synthesized view quality.

TABLE IV
RD PERFORMANCE OF TEXTURE ONLY FVSP BASED ON HTM5.1

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	0.0%	1.8%	0.5%	0.5%	0.6%	104.7%	169.0%
Kendo	0.0%	0.1%	0.6%	0.3%	0.3%	0.5%	105.0%	180.8%
Newspaper_C	0.0%	1.4%	1.6%	0.7%	0.8%	0.6%	106.5%	156.5%
GT_Fly	0.0%	-7.3%	-7.8%	-1.9%	-1.9%	-1.5%	103.0%	159.5%
Poznan_Hall2	0.0%	-0.7%	0.3%	0.1%	0.1%	0.3%	104.4%	168.2%
Poznan_Stree	0.0%	-2.9%	-2.7%	-0.8%	-0.8%	-0.5%	104.2%	164.4%
Undo_Dancer	0.0%	-14.2%	-13.4%	-3.8%	-3.7%	-3.2%	105.1%	165.6%
1024x768	0.0%	0.5%	1.4%	0.5%	0.5%	0.6%	105.4%	168.8%
1920x1088	0.0%	-6.3%	-5.9%	-1.6%	-1.6%	-1.2%	104.2%	164.4%
average	0.0%	-3.4%	-2.8%	-0.7%	-0.7%	-0.5%	104.7%	166.3%

Next, we study the VSP usage for different test sequences. Fig. 9(a) illustrates the VSP usage for the test sequence PoznanStreet of size 1920×1088 for the dependent view 1 at the anchor frame (base view intra coded). And Fig. 9(b) illustrates the VSP usage for dependent view 2 at the anchor frame. It is observed that VSP is chosen in around 20%-30% area within the picture. Generally, VSP modes are more frequently chosen in the anchor frame than in the non-anchor frames. This suggests that although VSP is quite efficient in the anchor frame competing with translational block-based DCP, it is relatively less efficient in the non-anchor frames competing with both translational block-based MCP and DCP.

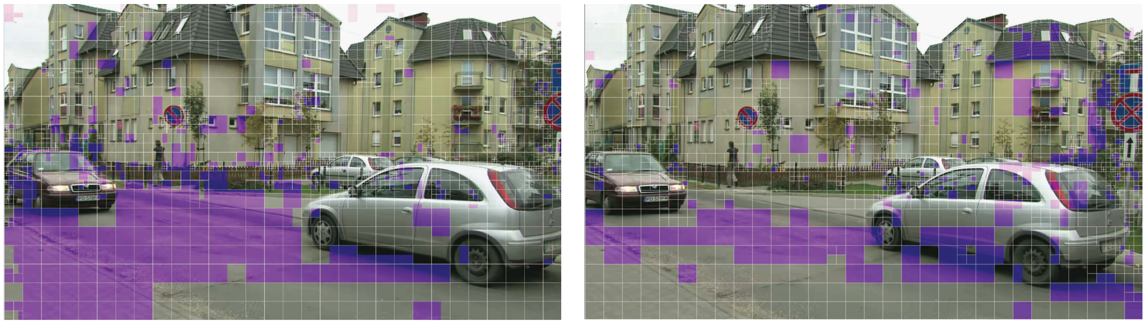
Another observation is that the VSP mode tends to be more frequently used in smooth regions than in the edge regions. The reason is that when the depth used in VSP is of low accuracy, the pixel correspondence is not reliable and thus the VSP predictor relying on the pixel correspondence tends to be of low quality, especially for the edge and occlusion regions, resulting in a large residue. Therefore, accurate depth is highly desirable in establishing a correct pixel correspondence in VSP. Among different video sequences, VSP modes are more likely to be chosen when the depth is obtained with high accuracy. For instance, for the sequence *UndoDancer*, the depth is obtained through the computer animation method, which possesses high accuracy and accurate alignment with the texture content. With the accurate depth, *UndoDancer* has the most frequent usage 45% of VSP modes among all the test sequences and it benefits the most from VSP modes with 14.2% bitrate reduction for dependent view 1.

Although the frame based FVSP improves the coding efficiency of 3D-HEVC, it entails the hole filling process, which is data dependent and thus irregular for data access and hardware implementation. In addition, FVSP requires the entire synthesized picture to be generated at the decoder regardless of the usage of VSP mode for each block. In an extreme case, there is no block chosen as VSP while the entire synthesized frame is generated. This undoubtedly poses unnecessary computation complexity for the decoder as shown in Table III.

C. BVSP in 3D-HEVC

To reduce the FVSP complexity and support the texture first coding order of 3D-HEVC, BVSP is proposed in Section III-B and implemented on top of HTM5.1. As discussed earlier in Section III-B, BVSP is realized by incorporating a VSP merge candidate (referring to a synthetic block with a displacement vector (0,0)) in the merge candidate list.

Note that in HTM5.1, the merge list is constructed in a predefined order, e.g., Inter-view, A1, B1, B0, A0, B2 and temporal right down block RB (collocated if RB is not



(a) (b)

Fig. 9. The usage of Prediction Unit (PU) chosen as VSP for HTM5.1 with VSP Skip/Merge candidates. The shaded area represents the VSP PUs. The white square represents the Coding Unit (CU) partition. (a) The usage of VSP PUs for PoznanStreet dependent view 1, (b) The usage of VSP PUs for PoznanStreet dependent view 2.

TABLE V
RD PERFORMANCE OF BVSP WITH THE VSP MERGE CANDIDATE BASED ON HTM5.1.

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	-1.3%	-0.2%	-0.2%	-0.1%	-0.2%	111.2%	115.1%
Kendo	0.0%	-0.8%	-0.5%	-0.1%	-0.1%	-0.3%	108.9%	106.6%
Newspaper_C	0.0%	-0.3%	-0.2%	-0.1%	-0.1%	-0.2%	107.3%	107.8%
GT_Fly	0.0%	-8.3%	-7.9%	-2.1%	-1.9%	-1.5%	104.3%	107.5%
Poznan_Hall2	0.0%	-0.8%	-0.9%	-0.3%	-0.2%	-0.3%	100.3%	93.6%
Poznan_Street	0.0%	-2.5%	-2.1%	-0.6%	-0.5%	-0.5%	108.7%	103.1%
Undo_Dancer	0.0%	-12.2%	-10.4%	-3.3%	-3.0%	-2.4%	98.6%	99.9%
1024x768	0.0%	-0.8%	-0.3%	-0.1%	-0.1%	-0.2%	109.2%	109.8%
1920x1088	0.0%	-6.0%	-5.3%	-1.6%	-1.4%	-1.2%	103.0%	101.0%
average	0.0%	-3.8%	-3.2%	-1.0%	-0.8%	-0.8%	105.6%	104.8%

TABLE VI
RD PERFORMANCE OF BVSP WITH VSP MERGE CANDIDATE AND DEPTH-ASSISTED MVP BASED ON HTM5.1.

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	-1.5%	-0.9%	-0.4%	-0.3%	-0.5%	100.3%	103.0%
Kendo	0.0%	-1.6%	-1.9%	-0.6%	-0.5%	-0.6%	100.5%	101.9%
Newspaper_C	0.0%	-0.6%	-0.9%	-0.2%	-0.2%	-0.3%	102.1%	101.2%
GT_Fly	0.0%	-8.9%	-8.7%	-2.4%	-2.2%	-1.7%	96.4%	102.0%
Poznan_Hall2	0.0%	-0.4%	-2.8%	-0.6%	-0.5%	-0.6%	100.9%	101.1%
Poznan_Street	0.0%	-3.0%	-3.2%	-1.0%	-0.9%	-0.8%	100.3%	102.5%
Undo_Dancer	0.0%	-12.4%	-11.0%	-3.4%	-3.1%	-2.5%	105.3%	111.4%
1024x768	0.0%	-1.2%	-1.2%	-0.4%	-0.3%	-0.5%	101.0%	102.0%
1920x1088	0.0%	-6.2%	-6.4%	-1.8%	-1.7%	-1.4%	100.7%	104.3%
average	0.0%	-4.1%	-4.2%	-1.2%	-1.1%	-1.0%	100.8%	103.3%

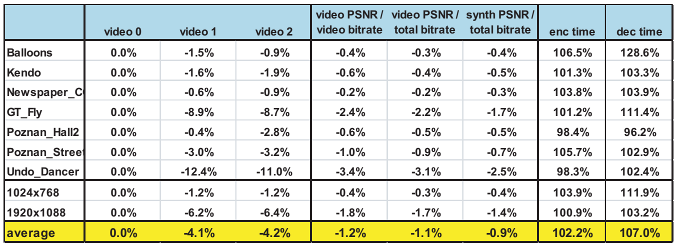
available). The insertion of the VSP candidate can be arranged at different positions in the list, resulting in different merge lists. Unary code is used to represent the selected index in the bitstream. Typically, placing a candidate at the beginning of the list would use shorter codewords, and vice versa. In our experiments, we find that a higher coding gain is obtained for sequences with a higher VSP usage (e.g. *UndoDancer*), and vice versa. Also, setting the VSP candidate right after B2 provides the most coding gain on average. Therefore, we choose the VSP candidate position right after B2 in our proposed BVSP scheme. Although the VSP candidate is added in the list, it should be noted that the maximum number of candidates is kept unchanged as six in our proposed design. In other words, if the number of available candidates exceeds six, only the first six candidates are used. If the number of available candidates do not reach six, generated candidates or (0,0) are filled for the remaining candidates. The design of fixing the maximum candidate size would remove the parsing dependency as no list construction is needed during the syntax parsing process. Table V shows that the proposed BVSP scheme with the inserted VSP merge candidate provides 1.0% bitrate reduction on average for coded video, 0.8% bitrate reduction on average for coded video vs. total bitrate, and 0.8% bitrate reduction on average for synthesized views.

Recall that in Section IV, a depth block from the current view is used to derive the proposed depth-assisted MVP. However, the current view depth is not available when coding the current view texture in the texture first coding order of 3D-HEVC. To solve this problem, in our implementation, the depth block is estimated similarly as done in BVSP Step 1, where a neighboring block disparity vector is used to locate a reference depth block in the base view and the located depth block is assumed as a good approximation of the current depth block. Given the depth block, the per pixel depth value is converted to the per pixel disparity using a look-up table initialized with the given camera parameters. And the converted per pixel disparity is used as MVP, which we call it depth-assisted MVP in the following. In case the block is using the VSP mode, the depth-assisted MVP is not only used at per pixel disparity vector predictor but also used as the compensation vector. In case the block is using block-based translational DCP, a block-based disparity vector is generated from the per pixel depth-assisted MVP, where we select the disparity vector of the maximum depth value as the representative disparity vector for the current block. And the generated disparity vector replaces the neighboring block disparity vector (with which the depth block is fetched) as the MVP for the current block. The proposed depth-assisted MVP scheme is also implemented on top of HTM5.1. When the depth-assisted MVP is combined with the BVSP scheme, additional 0.2% bitrate reduction is achieved for coded video and synthesized video respectively as shown in Table VI.

Note that the coding gain in Table VI is obtained by applying the VSP scheme to both texture and depth. To investigate coding gain contribution from each component, Table VII shows 0.9% bitrate reduction achieved for synthesized video when the proposed VSP scheme is applied to texture only. Comparing Table VII and VI, we conclude that applying VSP to the texture component only achieves the majority coding gain.

TABLE VII

RD PERFORMANCE OF BVSP WITH VSP MERGE CANDIDATE AND DEPTH-ASSISTED MVP APPLIED ON TEXTURE ONLY BASED ON HTM5.1.



	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time
Balloons	0.0%	-1.5%	-0.9%	-0.4%	-0.3%	-0.4%	106.5%	128.6%
Kendo	0.0%	-1.6%	-1.9%	-0.6%	-0.4%	-0.5%	101.3%	103.3%
Newspaper_C	0.0%	-0.6%	-0.9%	-0.2%	-0.2%	-0.3%	103.8%	103.9%
GT_Fly	0.0%	-8.9%	-8.7%	-2.4%	-2.2%	-1.7%	101.2%	111.4%
Poznan_Hall2	0.0%	-0.4%	-2.8%	-0.6%	-0.5%	-0.5%	98.4%	96.2%
Poznan_Stree	0.0%	-3.0%	-3.2%	-1.0%	-0.9%	-0.7%	105.7%	102.9%
Undo_Dancer	0.0%	-12.4%	-11.0%	-3.4%	-3.1%	-2.5%	98.3%	102.4%
1024x768	0.0%	-1.2%	-1.2%	-0.4%	-0.3%	-0.4%	103.9%	111.9%
1920x1088	0.0%	-6.2%	-6.4%	-1.8%	-1.7%	-1.4%	100.9%	103.2%
average	0.0%	-4.1%	-4.2%	-1.2%	-1.1%	-0.9%	102.2%	107.0%

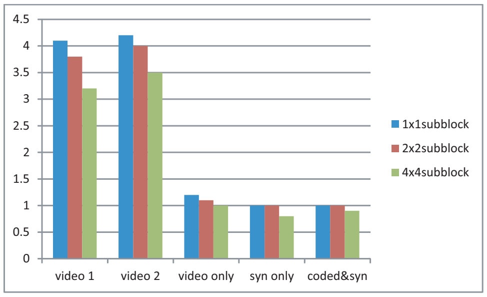


Fig. 10. Coding gain of BVSP with different sub-block sizes.(%)

Recall that in our proposed BVSP scheme, each pixel has its own disparity vector. In most cases, the disparity vectors within a block are similar because the pixels within the block are quite near to each other and often represent the same object with similar depth values. To reduce the number of disparity vectors for a block during compensation, a sub-block based disparity vector is proposed in our BVSP. For example, when 4×4 sub-block is used, for a 16×8 PU, eight disparity vectors are used during the compensation process. Typically when the sub-block size increases, the improvement brought by BVSP is reduced. In other words, reducing the disparity number per prediction unit would generally dilute the coding gain of VSP. Fig. 10 demonstrates the trend when constraining the sub-block size to 4x4, 2x2 and 1x1 in BVSP. It is found that although the coding gain is reduced from 1x1 to 4x4 sub-block BVSP, 4x4 sub-block BVSP maintains the majority gain. Thus the 4x4 sub-block BVSP scheme has been adopted into the 3D-HEVC standard [18].

Later, to further reduce the compensation complexity, the sub-block VSP in the latest 3D-HEVC (HTM 9.0) is now only operated in 4×8 or 8×4 mode [25]. It is worth mentioning that the proposed sub-block VSP in [25] has negligible coding loss compared with 4×4 VSP [18]. The BVSP scheme with depth-assisted MVP on HTM 9.0 shows similar performance as that on HTM 5.1 (1.3% bitrate reduction for video PSNR vs. video bitrate, 1.2% bitrate reduction for video PSNR vs. total bitrate, and 1.0% bitrate reduction for synthesis PSNR vs. total bitrate). In addition, to support IBP interview coding structure, we proposed a way to generalize the BVSP scheme by referring to different reference pictures according to the derived disparity from neighboring blocks [35], which was adopted in the 3D-HEVC standard.

D. Complexity

On the complexity, the proposed BVSP scheme has comparable encoder and decoder complexity with the anchor in terms of run time. The proposed FVSP for 3D-HEVC has comparable encoder complexity but relatively large decoder complexity: an additional 90.9% decoding time when FVSP is applied to both texture and depth, and an additional 66.3% decoding time when FVSP is applied to texture only. With regards to 3D-AVC, since it is based on BVSP, the proposed VSP Skip and Direct modes have comparable encoder and decoder complexity as expected. In summary, FVSP-based and BVSP-based VSP schemes have similar coding gain; however, the BVSP-based VSP scheme has much lower decoder complexity. Therefore, the BVSP design strikes a better trade-off between the coding gain and complexity.

VI. CONCLUSION

In this paper, we considered several 3D coding designs that utilize view synthesis prediction to improve the coding efficiency of multiview plus depth data formats. A novel backward-VSP scheme that uses a derived disparity vector was proposed, and efficient signalling methods have been presented. Our proposed approaches have been adopted into 3D extensions of both the AVC and HEVC standards. Additionally, we derived mathematical expressions of the RD property associated with VSP in an RD optimized video coding framework, which motivated the development of a novel depth-assisted motion vector prediction scheme. Extensive experiments have been conducted to demonstrate the notable coding efficiency gains of the different VSP schemes in different codec designs.

ACKNOWLEDGMENT

The authors would like to thank Lu Fang, Jingjing Dai, Chao Pang, and Xingyu Zhang for their constructive discussions.

REFERENCES

[1] http://www.imax.com/about/experience/3d/.

[2] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626 –642, Apr. 2011.

[3] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging mvc standard for 3d video services," *EURASIP J. Appl. Signal Process.*, vol. 2009, pp. 8:1–8:13, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1155/2009/786015

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

[5] ——, "Multi-view video plus depth representation and coding," *Image Processing, IEEE International Conference on*, vol. 1, pp. 201–204, Oct. 2007.

[6] K. Muller, P. Merkle, and T. Wiegand, "3-d video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[7] S. Liu, P. Lai, D. Tian, and C. W. Chen, "New depth coding techniques with utilization of corresponding video," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 551–561, 2011.

[8] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pp. 245–248, May 2008.

[9] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Muller, H. F. Rhee, G. Tech, M. Winken, and T. Wiegand, “Description of 3d video technology proposal by fraunhofer hhi (hevc compatible; configuration b),” in *MPEG Meeting - ISO/IEC JTC1/SC29/WG11, Doc. MPEG11/M22571, Geneva, CH*, Dec. 2011.

[10] K. Muller, P. Merkle, G. Tech, and T. Wiegand, “3d video coding with depth modeling modes and view synthesis optimization,” *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pp. 1–4, Dec. 2012.

[11] Y.-W. Chen, J.-L. Lin, Y.-W. Huang, and S. Lei, “3D-CE3.h results on removal of parsing dependency and picture buffers for motion parameter inheritance,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-C0137*, Geneva, CH, Jan. 2013.

[12] E. Martinian, A. Behrens, J. Xin, and A. Vetro, “View Synthesis for Multiview Video Compression,” in *Picture Coding Symposium (PCS)*, Apr. 2006.

[13] S. Yea and A. Vetro, “View synthesis prediction for multiview video coding,” *Image Commun.*, vol. 24, no. 1-2, pp. 89–100, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.image.2008.10.007

[14] S. Shimizu, H. Kimata, and Y. Ohtani, “Adaptive appearance compensated view synthesis prediction for multiview video coding,” *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 2949–2952, Nov. 2009.

[15] S. Yea and A. Vetro, “Rd-optimized view synthesis prediction for multiview video coding,” *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, pp. 209–212, Oct. 2007.

[16] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, “View scalable multiview video coding using 3-d warping with depth map,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[17] C. Lee and Y.-S. Ho, “A framework of 3d video coding using view synthesis prediction,” in *Picture Coding Symposium (PCS), 2012*, May 2012, pp. 9–12.

[18] D. Tian, F. Zou, and A. Vetro, “CE1.h: Backward View Synthesis Prediction using Neighboring Blocks,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-C0152*, Geneva, CH, Jan. 2013.

[19] L. Zhang, Y. Chen, and M. Karczewicz, “3D-CE5.h: Disparity vector generation results,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-A0097*, Stockholm, SE, Jul. 2012.

[20] X. Zhao, Y. Chen, L. Zhang, J. Kang, Y.-K. Wang, R. Joshi, and M. Karczewicz, “CE7: MB-level NBDV for 3D-AVC,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-E0136*, Vienna, AT, Jul. 2012.

[21] W. Su, D. Rusanovskyy, and M. M. Hannuksela, “3DV-CE1.a: Block-based View Synthesis Prediction for 3DV-ATM,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-A0107*, Stockholm, SE, Jul. 2012.

[22] C.-L. Wu, Y.-L. Chang, Y.-P. Tsai, and S. Lei, “3D-CE1.a related: Interview Skip mode with sub-partition scheme,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-A0050*, Stockholm, SE, Jul. 2012.

[23] F. Zou, D. Tian, A. Vetro, S. Shimizu, S. Sugimoto, and H. Kimata, “CE1.h: Results on View Synthesis Prediction,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-B0102*, Shanghai, CN, Oct. 2012.

[24] S. Shimizu, S. Sugimoto, H. Kimata, D. Tian, F. Zou, and A. Vetro, “3D-CE1.h Results on View Synthesis Prediction,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-A0018*, Stockholm, SE, Jul. 2012.

[25] S. Shimizu and S. Sugimoto, “3D-CE1.h: Adaptive block partitioning for VSP,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-E0207*, Vienna, AT, Jul. 2013.

[26] J. Sung, M. Koo, and S. Yea, “3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D Video Coding,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT2-A0126*, Stockholm, SE, Jul. 2012.

[27] T. Kim, J. Nam, and S. Yea, “CE1.h related : BVSP mode inheritance,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-D0092*, Incheon, KR, Apr. 2013.

[28] D. Tian, D. Graziosi, and A. Vetro, “3D-AVC-CE1 Results on VSP Skip/Direct Mode by Mitsubishi,” *ISO/IEC JTC 1/SC 29/WG 11, MPEG2011/M23915*, San Jose, USA, Feb. 2012.

[29] Y.-L. Chang, C.-L. Wu, Y.-P. Tsai, and S. Lei, “3D-CE1.h: Depth-oriented neighboring block disparity vector (DoNBDV) with virtual depth retrieval,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-C0131*, Geneva, CH, Jan. 2013.

[30] D. Rusanovskyy, K. Miller, and A. Vetro, “Common Test Conditions of 3DV Core Experiments,” *ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCT3V-B1110*, Shanghai, CN, Oct. 2012.

[31] http://mpeg3dv.research.nokia.com/svn/mpeg3dv/tags/3DV ATMv6.0.

[32] G. Bjontegaard, “Calculation of Average PSNR Differences between RD curves,” *ITU-T SC16/SG16 VCEG, VCEG-m33,Texas, Austin, USA*, Apr. 2001.

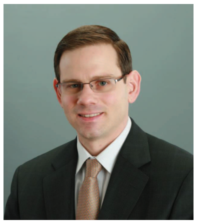
[33] J. Y. Lee, J. Lee, and D. S. Park, “3D-CE1.a results on a context-based adaptive skip flag positioning method by Samsung,” *ISO/IEC JTC/SC29/WG11 MPEG document m24819*, Geneva, CH, Apr. 2012.

[34] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM 5.1.

[35] F. Zou, D. Tian, A. Vetro, and H. Sun, “3D-CE1.h: On reference view selection in NBDV and VSP,” *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT3V-D0166*, Incheon, KR, Apr. 2013.

Feng Zou received the B.S. degree in electrical engineering from Harbin Institute of Technology, China, in 2004 and the Ph.D. degree in electronic and computer engineering in the Hong Kong University of Science and Technology (HKUST), Hong Kong SAR. While at HKUST, he was supervised by Prof. Oscar C. Au. From June 2012 to June 2013, he interned in Mitsubishi Electric Research Labs (MERL) in Cambridge, USA. Now, he works at Qualcomm as a senior engineer in San Diego, and his current research interests include intra prediction, transform, quantization designs, 3D video coding and screen content coding. He is actively contributing proposals in the standardization of HEVC under the ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), JCT3V and Range extension of HEVC and holds several video coding related patents.

Dong Tian received the Ph.D. degree in electrical engineering from Beijing University of Technology in 2001, and M.Eng and B.Eng degrees in automation from University of Science and Technology of China (USTC) in 1998 and 1995, respectively. He is currently a Senior Principal Member Research Staff in Multimedia Group of Mitsubishi Electric Research Labs (MERL) at Cambridge, MA. His current responsibilities are conducting researches on image/video coding and processing and 3D images, including 3D-HEVC, a recent extension of HEVC to support advanced 3D video data format. Prior to joining MERL in 2010, he worked with Thomson Corporate Research at Princeton, NJ for about four years, where he was devoted to H.264/AVC encoder design and optimizations and 3D video processing, especially made contributions to H.264/AVC extensions to support Multiview Video Coding (MVC) and later on 3D Video (3DV) within MPEG. Before that, he had been working as a visiting researcher at Tampere University of Technology in Finland from 2002 to 2005 on the development of H.264/AVC funded by Nokia Research Centre. Dr. Tian is a member of IEEE.



Anthony Vetro (S‘92-M‘96-SM‘04-F‘11) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY. He joined Mitsubishi Electric Research Labs, Cambridge, MA, in 1996, where he is currently a Group Manager responsible for research and standardization on video coding, as well as work on display processing, information security, sensing technologies, and speech/audio processing. He has published more than 150 papers in these areas. He has also been an active member of the ISO/IEC and ITU-T standardization committees on video coding for many years, where he has served as an ad-hoc group chair and editor for several projects and specifications. He was a key contributor to the Multiview Video Coding extension of the H.264/MPEG-4 AVC standard, and current serves as Head of the U.S. delegation to MPEG. Dr. Vetro is also active in various IEEE conferences, technical committees, and editorial boards. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, and as a member of the Editorial Boards of IEEE MultiMedia and IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He served as Chair of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and on the steering committees for ICME and the IEEE TRANSACTIONS ON MULTIMEDIA. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2010-2013) and IEEE Signal Processing Magazine (2006-2007) and, and later served as a member of the Editorial Board (2009-2011). He also served as a member of the Publications Committee of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS (2002-2008). He has also received several awards for his work on transcoding, including the 2003 IEEE Circuits and Systems CSVT Transactions Best Paper Award. He is a Fellow of IEEE.

Huifang Sun graduated from Harbin Engineering Institute, China in 1967, and received the Ph.D. from University of Ottawa, Canada in 1986. He joined Electrical Engineering Department of Fairleigh Dickinson University in 1986 and was promoted to an Associate Professor before moved to Sarnoff Corporation in 1990. He joined Sarnoff Lab as a member of technical staff and was promoted to Technology Leader of Digital Video Communication later. In 1995, he joined Mitsubishi Electric Research Laboratories (MERL) as Senior Principal Technical Staff and was promoted as Vice President and Fellow of MERL and in 2003 and now is a MERL Fellow. His research interests include digital video/image compression and digital communication. He has coauthored two books and more than 150 Journal/and conference papers. He holds 65 US patents. He received Technical Achievement Award in 1994 at Sarnoff Lab. He received the best paper award of 1992 IEEE Transaction on Consumer Electronics, the best paper award of 1996 ICCE and the best paper award of 2003 IEEE Transaction on CSVT. He was an Associate Editor for IEEE Transaction on Circuits and Systems for Video Technology and the Chair of Visual Processing Technical Committee of IEEE Circuits and System Society. He is an IEEE Life Fellow.

Oscar C. Au received his B.A.Sc. from Univ. of Toronto in 1986, his M.A. and Ph.D. from Princeton Univ. in 1988 and 1991 respectively. After being a postdoc in Princeton for 1 year, he joined Hong Kong Univ. of Science and Technology (HKUST) as an Assistant Professor in 1992. He is/was a Professor of Dept. of Electronic and Computer Engineering, Director of Multimedia Technology Research Center, and Director of Computer Engineering in HKUST. He is a core member of the State Key Lab on Advanced Displays and Optoelectronic Technology.

His main research contributions are on video/image coding and processing, watermarking/light weight encryption, speech/audio processing. Research topics include fast motion estimation for H.261/3/4/5, MPEG-1/2/4, and AVS, optimal and fast sub-optimal rate control, mode decision, transcoding, denoising, deinterlacing, post-processing, multi-view coding, view interpolation, depth estimation, 3DTV, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000, HDR imaging, compressive sensing, halftone image data hiding, GPU-processing, software-hardware co-design, etc. He has published 60+ technical journal papers, 350+ conference papers, 3 book chapters, and 70+ contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His lightweight encryption and error resilience algorithms are accepted into the China AVS standard. He was Chair of Screen Content Coding AdHoc Group in JCTVC for HEVC. He has 20+ granted US patents and is applying for 70+ more on his signal processing techniques. He has performed forensic investigation and stood as an expert witness in Hong Kong courts many times.

Dr. Au is a Fellow of IEEE and HKIE. He is/was Associate Editors of several IEEE journals (TCSVT, TIP, TCAS1, SPL, JSTSP, SigView) and non-IEEE journals (JVCIR, JSPS, TSIP, JMM, JFI, and SWJ). He is guest editor of some special issues in JSTSP and TCSVT. He is/was BoG member and Vice President - Technical Activity of APSIPA. He is/was Chair of 3 technical committees: IEEE CAS MSA TC, IEEE SPS MMSP TC, and APSIPA IVM TC. He is a member of 5 other TCs: IEEE CAS VSPC TC, DSP TC, IEEE SPS IVMSP TC, IFS TC, and IEEE ComSoc MMC TC. He served on 2 steering committees: IEEE TMM, and IEEE ICME. He also served on organizing committee of many conferences including ISCAS 1997, ICASSP 2003, ISO/IEC 71st MPEG in Jan 2005, ICIP 2010, etc. He was/will be General Chair of several conferences: PCM 2007, ICME 2010, PV 2010, MMSP 2015, APSIPA ASC 2015, and ICME 2017. He won 5 best paper awards: SiPS 2007, PCM 2007, MMSP 2012, ICIP 2013, and MMSP 2013. He was IEEE Distinguished Lecturer (DL) in 2009 and 2010, APSIPA DL in 2013 and 2014, and has been keynote speaker multiple times.

Shinya Shimizu received the B.Eng. and M.Info. degrees in social informatics from Kyoto University, Japan, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from Nagoya University, Japan, in 2012. He joined Nippon Telegraph and Telephone Corporation (NTT) in 2004 and has been engaged in R&D of 3D video coding algorithms and standardizations in MPEG and ITU-T. His research interests also includes signal processing for free viewpoint video, computer vision, and computational photography. He is currently a Research Engineer at Visual Media Project in NTT Media Intelligence Laboratories. Dr. Shimizu is a member of Institute of Electronics, Information and Communication Engineers of Japan (IEICE).