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

View Synthesis Prediction for Multiview Video Coding

Sehoon Yea, Anthony Vetro

TR2009-009 April 2009

Abstract

We propose a rate-distortion-optimized framework that incorporates view synthesis for improved prediction in multiview video coding. In the proposed scheme, auxiliary information, including depth data, is encoded and used at the decoder to generate the view synthesis prediction data. The proposed method employs optimal mode decision including view synthesis prediction, and sub-pixel reference matching to improve prediction accuracy of the view synthesis prediction. Novel variants of the skip and direct modes are also presented, which infer the depth and correction vector information from neighboring blocks in a synthesized reference picture to reduce the bits needed for the view synthesis prediction mode. We demonstrate two multiview video coding scenarios in which view synthesis prediction is employed. In the first scenario, the goal is to improved the coding efficiency of multiview video where block-based depths and correction vectors are encoded by CABAC in a lossless manner on a macroblock basis. A variable block-size depth/motion search algorithm is described. Experimental results demonstrate that view synthesis prediction does provide some coding gains when combined with disparity-compensated prediction. In the second scenario, the goal is to use view synthesis prediction for reducing rate overhead incurred by transmitting depth maps for improved support of 3DTV and free-viewpoint video applications. It is assumed that the complete depth map for each view is encoded separately from the multiview video and used at the receiver to generate intermediate views. We utilize this information for view synthesis prediction to improve overall coding efficiency. Experimental results show that the rate overhead incurred by coding depth maps of varying quality could be offset by utilizing the proposed view synthesis prediction techniques to reduce the bitrate required for coding multiview video.

Signal Processing: *Image Communication* 24 (2009) 89–100

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

View synthesis prediction for multiview video coding

Sehoon Yea *, Anthony Vetro

MERL – Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

ARTICLE INFO

Article history:
Received 8 October 2008
Accepted 19 October 2008

Keywords:
Multiview video coding
View synthesis
Prediction
Depth
3DTV
Free-viewpoint video

ABSTRACT

We propose a rate-distortion-optimized framework that incorporates view synthesis for improved prediction in multiview video coding. In the proposed scheme, auxiliary information, including depth data, is encoded and used at the decoder to generate the view synthesis prediction data. The proposed method employs optimal mode decision including view synthesis prediction, and sub-pixel reference matching to improve prediction accuracy of the view synthesis prediction. Novel variants of the skip and direct modes are also presented, which infer the depth and correction vector information from neighboring blocks in a synthesized reference picture to reduce the bits needed for the view synthesis prediction mode. We demonstrate two multiview video coding scenarios in which view synthesis prediction is employed. In the first scenario, the goal is to improve the coding efficiency of multiview video where block-based depths and correction vectors are encoded by CABAC in a lossless manner on a macroblock basis. A variable block-size depth/motion search algorithm is described. Experimental results demonstrate that view synthesis prediction does provide some coding gains when combined with disparity-compensated prediction. In the second scenario, the goal is to use view synthesis prediction for reducing rate overhead incurred by transmitting depth maps for improved support of 3DTV and free-viewpoint video applications. It is assumed that the complete depth map for each view is encoded separately from the multiview video and used at the receiver to generate intermediate views. We utilize this information for view synthesis prediction to improve overall coding efficiency. Experimental results show that the rate overhead incurred by coding depth maps of varying quality could be offset by utilizing the proposed view synthesis prediction techniques to reduce the bitrate required for coding multiview video.

1. Introduction

Emerging applications in multiview video such as free-viewpoint TV (FTV) [13,5,6], 3D displays [2], and high-performance imaging [14] require dramatic increase in bandwidth for their dissemination and make compression ever more important. Consequently, there are growing interests in coding techniques that take advantage of the correlation among neighboring camera views. In response to such needs and interests, recent multiview coding standardization activities by MPEG/JVT have been focused on developing generic coding toolsets geared mainly toward compression efficiency improvement by capitalizing on the inter-view correlation existing among views [12].

Disparity-compensated prediction (DCP) is a well-known technique for exploiting the redundancy between different views. This prediction mode provides gains when the temporal correlation is lower than the spatial correlation, e.g., due to occlusions, objects entering or leaving the scene, or fast motion. Martinian et al. [8] first proposed the use of view synthesis prediction (VSP) as an additional method of prediction in which a synthesized picture is generated from neighboring views using depth

* Corresponding author. Tel.: +1 617 621 7509.
E-mail addresses: yea@merl.com (S. Yea), avetro@merl.com (A. Vetro).
URL: http://www.merl.com/people/?user=yea (S. Yea).

0923-5965/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2008.10.007

information and used as a reference for prediction. This prediction mode is expected to be complementary to disparity compensation due to the existence of non-translational motion between camera views and provide gains when the camera parameters and estimated depth of the scene are accurate enough to provide high-quality synthetic views. A related VSP scheme by Shimizu et al. [11] proposes to encode view-dependent geometry information that enables a VSP.

In this paper, we first incorporate view synthesis into the block-based rate-distortion (RD)-optimization framework and describe a joint search algorithm for depth and correction vectors that are needed for high-quality view synthesis [15]. We encode this side information using CABAC and the rate overhead for this prediction mode is included in the final experimental results. We also introduce a novel extension of the skip and direct coding modes with respect to synthetic reference pictures. With these methods, we are able to infer the side information, thereby saving bits to generate the view synthesis reference data for a block, while maintaining prediction efficiency. Experimental results demonstrate VSP brings about additional coding gain when combined with disparity-compensated view prediction in multiview video coding.

Next, we demonstrate that the proposed VSP can also be a useful coding tool when generation of intermediate views is required at the receiver. For instance, to support auto-stereoscopic displays that require a higher number of views than transmitted, it is necessary to generate additional intermediate views. Intermediate views must also be generated in free viewpoint navigation scenarios. Given a discrete number of views with sufficient overlap, one synthesizes arbitrary intermediate views of interest using camera geometry and depth information; this scenario is commonly referred to as FTV.

From the coding perspective, this added requirement of generating intermediate views poses a challenging problem of efficiently compressing not only the multiview video itself, but also the associated depth maps of the scene. Since the requirement on the fidelity of the encoded depth maps will be often dictated by the expected rendering quality at the receiver side, it could imply a huge rate overhead in terms of coding and transmission. Therefore, it is desirable to be able to capitalize on the similarity between the multiview video and its associated depth maps for improved overall coding efficiency. In this context, we propose the (re-)use of encoded depth maps available both at the encoder and the decoder to improve coding efficiency of multiview video. In other words, the VSP technique utilizes the depth information that is already being encoded and transmitted; the depth is being made available to the receiver primarily for rendering purposes, but we utilize it for more efficient encoding. It is shown that the rate overhead incurred by coding high-quality depth maps can be offset by reducing the rate for coding multiview (texture) video with the proposed VSP technique. The results, however, also indicate that the amount of such rate reduction is not necessarily proportional to high rate overheads, e.g., when smaller quantization parameters (QPs) or sub-sampling ratios are used.

The rest of this paper is organized as follows. An overview of VSP is given in Section 2. In Section 3, we describe the RD-optimization framework including VSP. We also introduce the synthetic skip and direct modes that are natural extensions of the H.264/AVC standard. In Section 4, we discuss various issues related to generating and using side information such as depth and correction vectors. We then discuss issues and techniques for encoding the generated side information in Section 5. Experimental results are provided in Section 6 followed by concluding remarks in Section 7.

2. Overview of VSP

DCP typically utilizes a block-based disparity vector that provides the best matching reference position between a block in the current (predicted) view and reference view. In contrast, VSP attempts to utilize knowledge of the scene characteristics, including scene depth and camera parameters, to generate block-based reference data used for prediction. Fig. 1 illustrates a case where, in contrast with DCP which always maps every pixel in a rectangular MB in the predicted view to the corresponding pixel in the reference view displaced by the same amount using a single disparity vector, VSP maps to the matching pixels displaced not necessarily by the same amount and hence potentially provides better prediction when the matching area in the reference view

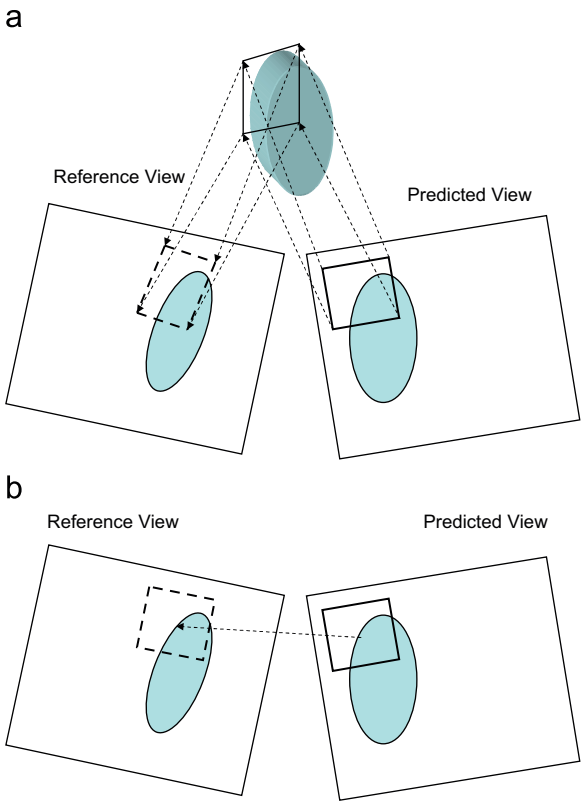

Fig. 1. Disparity compensated prediction (DCP, b) vs. view synthesis prediction (VSP, a).

is of non-rectangular shape or the correspondence between the views is non-translational. This could be the case even if a single depth value is used per MB and in fact is one of the key advantages of VSP over DCP. Of course, how significant this would be and under what precise conditions depend on a few other factors such as the MB size and the exact camera arrangement. Another advantage of VSP would be that it can have an arbitrarily large search range as it uses the camera parameters to locate the potential match. Finally, side information coding cost could be saved as depth is a scalar while disparity is a vector in general.

To obtain a synthesized reference picture, one needs to find the pixel intensity prediction $I'[c,t,x,y]$ for camera c ('predicted view') at time t for each pixel (x,y) of the current block to be predicted. We first apply the well-known pinhole camera model to project the pixel location (x,y) into world coordinates $[u,v,w]$ via

$$[u,v,w]^{\mathrm{T}} = R(c)\cdot A^{-1}(c)\cdot [x,y,1]^{\mathrm{T}}\cdot D[c,t,x,y] + T(c), \quad (1)$$

where D is the depth defined with respect to the optical center of the camera c and A, R and T are camera parameters [8]. Next, the world coordinates are mapped into the target coordinates $[x',y',z']$ of the frame in camera c' ('reference view') which we wish to predict from

$$[x',y',z']^{\mathrm{T}} = A(c')\cdot R^{-1}(c')\cdot [u,v,w]^{\mathrm{T}} - T(c'). \quad (2)$$

Then the intensity for pixel location (x,y) in the synthesized frame is given as $I'[c,t,x,y] = I[c',t,x'/z',y'/z']$. Finding the best depth D that maps (x,y) into (x',y') corresponds to the process of *sub-pixel matching* illustrated in Fig. 2. On the other hand, the process of using the best D to synthesize $I'[c,t,x,y]$ is labeled as *synthesis by warping* in Fig. 2. To further improve the performance of VSP, a synthesis correction vector (C_x,C_y) [9] could be introduced as illustrated in Fig. 2. Specifically, we replace (1) with the following:

$$[u,v,w]^{\mathrm{T}} = R(c)\cdot A^{-1}(c)\cdot [x+C_x,y+C_y,1]^{\mathrm{T}} \times D[c,t,x,y] + T(c). \quad (3)$$

We would like to mention in passing that a single depth (and optionally a correction vector) will be used for all the pixels in a macroblock in our proposed framework in order to strike a reasonable balance between the quality and the encoding rate of these information as well as naturally align with the traditional MB-based video coding standards such as H.264/AVC. The idea of using correction vectors as well as the required accuracies of depth values and correction vectors are studied further in Section 4.

3. RD-optimized framework

3.1. RD-optimal mode decision

In our previous work [9], we proposed a reference picture management scheme that allows the use of prediction in other views in the context of H.264/AVC without changing the lower layer syntax. This is achieved by placing reference pictures from neighboring views into a reference picture list with a given index. Then, disparity vectors are easily computed from inter-view reference pictures in the same way that motion vectors are computed from temporal reference pictures. This concept is easily extended to also accommodate prediction from view synthesis reference pictures. In the following, we present an RD-optimization framework that incorporates VSP and method for performing mode decision.

To describe the RD framework, we use MB to refer to different macroblock and sub-macroblocks partitions from 16×16 to 8×8. We define the cost of performing a motion or disparity compensated prediction for a given

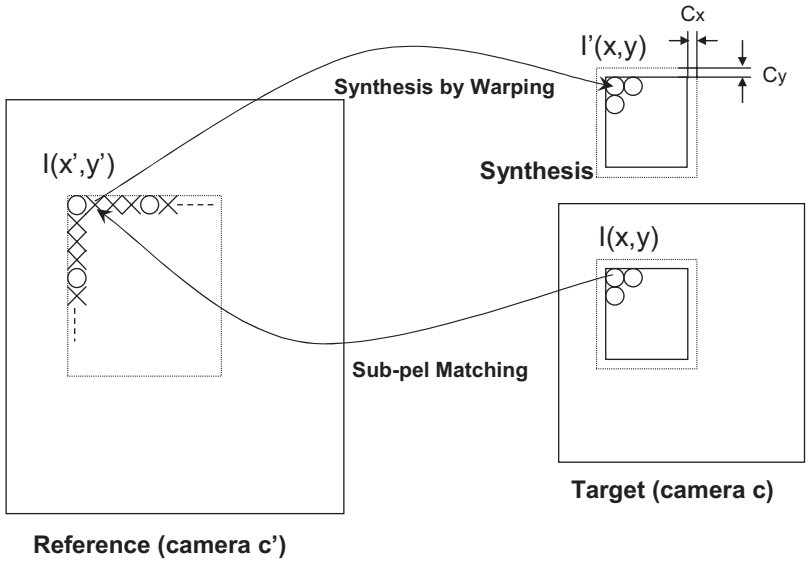

Fig. 2. Illustration of sub-pixel reference matching and correction vectors.

mb_type as

$$J_{motion}(\vec{m}, l_m|\text{mb\_type}) = \sum_{X\in\Phi} |X - X_p(\vec{m}, l_m)| + \lambda \cdot (R_m + R_{l_m}), \tag{4}$$

where $\vec{m}$ denotes a motion vector per MB with respect to the reference picture index l_m, R_m and R_{l_m} denote the bits for coding 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 Φ and its prediction, respectively. Similarly, the cost of performing a VSP is given by

$$J_{depth}(d, \vec{m}_c, l_d|\text{mb\_type}) = \sum_{X\in\Phi} |X - X_p(d, \vec{m}_c, l_d)| + \lambda \cdot (R_d + R_{m_c} + R_{l_d}), \tag{5}$$

where $(d, \vec{m}_c)$ denotes the depth/correction vector pair per MB with respect to a synthetic reference picture index l_d, and R_d, R_{m_c} and R_{l_d} denote the bits for coding the depth, correction vector and reference picture index, respectively. Then, either a motion vector, a disparity vector or a depth/correction vector pair that minimizes the following quantity J is chosen along with the corresponding reference frame index as the best inter-frame prediction candidate for each mb_type.

$$J = \min(J_{motion}, J_{depth}). \tag{6}$$

Following the above best candidate search for each mb_type, a mode decision is made in order to choose the mb_type (including intra-prediction modes also as candidates) that minimizes the Lagrangian cost function defined as

$$J_{mode}(\text{mb\_type}|\lambda_{mode}) = \sum_{X\in\Phi} (X - X_p)^2 + \lambda_{mode} \cdot (R_{side} + R_{res}), \tag{7}$$

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 either the depth/correction vector pair or the motion vector depending on the type of the reference picture.

Consider the case in which block-based depths as needed in the above formulation are encoded on a macroblock-basis as an integral part of the multiview video bitstream. In this case, VSP will only be chosen as the optimal mode when the RD cost is favorable compared to temporal, disparity compensated or intra prediction. In contrast, when the depth is assumed to be transmitted for intermediate view generation purposes as in FTV [5,6], it is then possible to discount the rate for coding depth in the above formulation. This creates a more favorable condition for VSP and the likelihood of using VSP under this assumption will increase. For example, without the depth coding rate penalty, $R_d = 0$ and $R_{m_c} = 0$ in (5) and R_{side} does not include the depth coding cost in (7). If the resulting cost for VSP in this case is smaller than the cost of other prediction modes, such as a temporal prediction with the associated motion vector coding cost, then the use of VSP is considered optimal in the RD sense. This is only true since the coded depth information is considered a fixed overhead. In Sections 5 and 6, we will discuss and demonstrate the uses of VSP in these two different scenarios. In the sequel, 'with depth penalty' will refer to the former case while 'without depth penalty' will refer to the latter.

3.2. Synthetic skip/direct modes

In conventional skip and direct coding modes in H.264/AVC, motion vector information and reference indices are derived from neighboring macroblocks. Considering inter-view prediction based on view synthesis, an analogous mode that derives depth and correction vector information from its neighboring macroblocks could be considered as well. We refer to this new coding mode as synthetic skip or direct mode.

In the conventional skip mode (P- or B-slice), no residual data are coded and the first entry (for P-skip) or the earliest entry among neighboring blocks (for B-skip) in

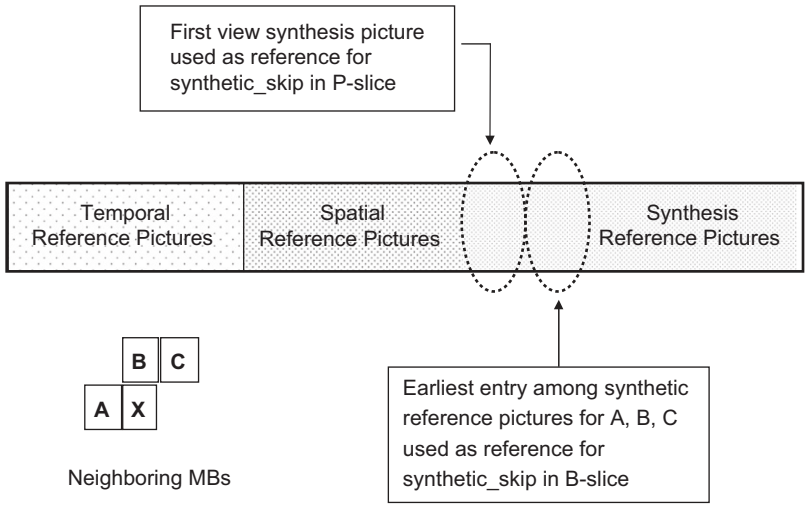

Fig. 3. Synthetic skip/direct modes.

the reference list is chosen as the reference to predict and derive information from. Since the method for reference picture list construction would simply append the view synthesis reference to the list, the reference picture for skip mode could never be a view synthesis picture with existing syntax. However, since the view synthesized picture may offer better quality compared to the disparity or motion compensated view, we consider a change to the slice data syntax and decoding process to allow for skip and direct modes based on the view synthesis reference.

To consider the skip mode with respect to a view synthesis reference, we introduce a synthetic skip mode that is signaled with modifications to the existing mb_skip_flag. Currently, when the existing mb_skip_flag is equal to 1, the macroblock is skipped, and when it is equal to 0, the macroblock is not skipped. With the proposed change, an additional bit is added in the case when mb_skip_flag is equal to 1 to distinguish the conventional skip mode from the new synthetic one. If the additional bit equals 1, this signals the synthetic skip, otherwise the conventional skip is used. When a synthetic skip mode is signaled, the first/earliest view synthesis reference picture in the reference picture list is chosen as the reference instead of the first/earliest entry in the reference picture list in the case of conventional skip as illustrated in Fig. 3.

The same signaling method is used for the direct-modes in B-slices, where residual information is present unless $\mathrm{cbp} = 0$. With these proposed synthetic skip and direct modes, it is possible to invoke the VSP modes with very little rate overhead for cases in which side information could be effectively inferred from neighboring blocks.

4. Side information generation

In this section, we discuss the issues associated with generating side information for VSP. The side information here refers to depth and synthesis correction vector. We discuss estimation and quantization of these quantities in the context of VSP.

4.1. Depth estimation algorithms

Estimating scene depth is one of the central themes of traditional computer vision research. However, there are many open problems and challenges in generating high-quality depth information of a scene and covering any of those issues is outside the scope of this paper. The intent here is to describe a simple block-based depth search algorithm mainly suited for VSP as well as to discuss some of the important issues in using the depth maps obtained by typical computer-vision-based approaches as an alternative.

4.1.1. Block search approach

We introduce a block-based depth search algorithm to find the optimal depth for each variable-size MB. Specifically, we define the minimum, maximum and incremental depth values d_{min}, d_{max} and d_{step}, respectively, where the minimum and maximum distances are defined with respect to the optical-center of the image plane of each camera from the objects. Note that these values are sequence/scene-dependent and could be available from the scene-capturing stage, or roughly estimated, e.g., by doing projection/re-projection of some test pixels between image frames from different views, as in [16]. Then, for each variable-size MB Φ in the frame we want to predict, we choose the optimal depth $D(c,t,\Phi)$ as follows:

$$D(c,t,\Phi) = \arg\min_{d\in\Delta} \sum_{(x,y)\in\Phi} \|I[c,t,x,y] - I[c',t,x',y']\|, \tag{8}$$

where $\Delta = \{d_{min}, d_{min}+d_{step}, d_{min}+2d_{step}, \ldots, d_{max}\}$, and the error measure, $\|I[c,t,x,y] - I[c',t,x',y']\|$, denotes the absolute difference of intensity values between the pixel at (x,y) in camera c at time t and the corresponding prediction pixel at (x',y') from camera c' as determined by (1) and (2) using the candidate depth d. It is noted that (8) will always produce the minimum prediction error, but the resulting depth map will generally be very noisy and not necessarily correspond to true depth values of the scene. Due to the noise and lack of correlation among neighboring depth values, the resulting data would also be quite difficult to compress. In order to overcome these drawbacks, we relax the minimization by introducing the depth coding rate R_d as a penalty term using a similar Lagrange multiplier λ as in (4) and (5):

$$D(c,t,\Phi) = \arg\min_{d\in\Delta} \sum_{(x,y)\in\Phi} \|I[c,t,x,y] - I[c',t,x',y']\| + \lambda \cdot R_d. \tag{9}$$

Recall from Section 2 that the correction vector is used to improve the view synthesis quality. In fact, we find the best combination of a depth and correction vector by searching over a small window (typically no larger than size 2×2) for the best correction vector that minimizes the SAD in (8) or (9) for each depth-value candidate d.

Other techniques to improve block-based depth search may also be considered in this framework. For instance, the benefits of a hierarchical search and inclusion of chrominance information as part of the cost function in (9) have been considered in [4].

4.1.2. Computer vision approaches

There are a vast amount of work in the computer vision literature addressing the problem of depth map estimation in the context of stereo as well as multiview vision. In our framework, we could consider using depth maps generated by such techniques, such as the one developed and used in [17]. There are a few points worth noting and discussing on this class of techniques in the context of multiview video compression. For one, the majority of stereo matching algorithms presume the use of rectified image pairs [10]. Since the epipolar lines align with the camera baseline after rectification, it greatly simplifies the disparity search and it is the very rationale behind the use of rectification. However, this implies one has to estimate the depth map in the rectified domain and then convert it back to the non-rectified domain for the later use [5]. This process might entail such issues as image distortion, clipping of boundary areas and multiview rectification. Another issue is that many of the well-known stereo matching algorithms [10] often fail to estimate reliable depth for real-world video sequences, such as the

MPEG/JVT multiview test set [12], possibly due to complex scene structures, wide baselines, illumination changes, etc. Furthermore, depth maps estimated on a frame-by-frame basis lack temporal consistency. Besides being difficult to compress, the temporal inconsistency also results in unnatural temporal flickering of rendered views; these are important considerations for the applications under consideration, and will be discussed further in Section 6.2. Finally, the prediction error is usually not the measure to be minimized when depth maps are estimated, hence true maps do not always produce the best prediction for coding.

4.2. Sub-pixel reference matching

Since the disparity of two corresponding pixels in different cameras is, in general, not given by an exact multiple of integers, the coordinates $[x', y', z']$ of the reference frame (as given by (2)) in camera c' which we wish to predict from does not always fall on the integer grid. Therefore, we propose to interpolate the sub-pixel positions in the reference frame. The matching algorithm then chooses the nearest sub-pixel reference point, thereby approximating the true disparity between the pixels more accurately. Fig. 2 illustrates this process. The same interpolation filters adopted for sub-pixel motion estimation in H.264/AVC were used in our implementation [1].

4.3. Depth quantization and correction vectors

Due to its inversely proportional relationship to disparity, it is well known that non-uniform quantization of depth could be beneficial in terms of producing higher rendering or prediction quality [3]. On the other hand, in addition to the sub-pixel reference matching discussed earlier, we found that the use of correction vectors (possibly with sub-pixel accuracy) as introduced in (3) helps to improve the quality of synthesized prediction.

Fig. 4 illustrates the effects of using different depth quantization schemes, as well as the effect of correction vectors, where the coordinates of the points represent the vertical and the horizontal components of the disparity vectors corresponding to depth values between 20 and 100 with the step size of 2, respectively, where the depth value is defined as the distance with respect to a camera coordinate and one unit in this example corresponds to about 5 cm. We used camera parameters for the break-dancers sequence for the mapping and the pixel coordinate of (300, 300) in view 2 is chosen as the current pixel for which the target coordinate in view 1 is to be found. Figs. 4(a) and (b) correspond to the cases without and with correction vectors when uniform quantization is used for depth, respectively. Similarly, Figs. 4(c) and (d) correspond to the cases without and with correction vectors when non-uniform quantization is used for depth, respectively. Twenty-five correction vectors with each component value ranging from −1 to 1 with the step size of 0.5 were used. As can be readily confirmed, the use of correction vectors leads to enlargement of the search area around the epipolar lines in the target image by adding extra search points on the parallel lines around it, while non-uniform quantization of depth spreads out the disparity search grid more evenly.

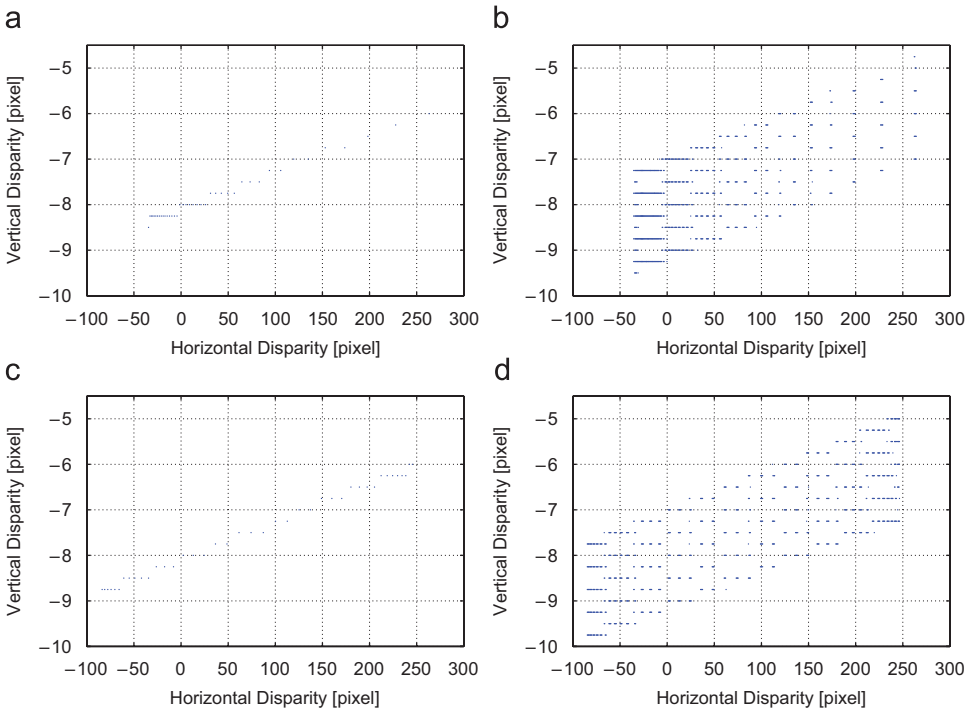

Fig. 4. Disparity search along the epipolar line. (a) Without correction vector, uniform depth quantization. (b) With correction vector, uniform depth quantization. (c) Without correction vector, non-uniform depth quantization. (d) With correction vector, non-uniform depth quantization.

It is known that using a small vertical component of disparity vector even after rectification improves stereo-matching quality [7] due to the inevitable parametric errors involved in rectification. In fact, the role of the correction vector is similar to that of such a vertical disparity component, the difference being that the search for the former is performed in the current view to be predicted as in (3) while that for the latter is done in the reference view to predict from. The design choice in the proposed scheme comes from the fact that having a single correction vector on the current MB might lead to multiple equivalent disparity 'adjustments' for the pixels in the current MB when the inter-view motion is not purely translational, which might be beneficial in a similar way VSP is over DCP. It might be of interest to evaluate other design choices (e.g., having a correction vector or a scalar displacement perpendicular to the nominal epipolar line on the reference view), but this is outside the scope of the current work.

Finally, note that the best combination of quantization schemes for depth and correction vector to synthesize a prediction for a target MB should be determined in the RD-optimal sense. In other words, both the quality of the synthesized prediction and the coding cost should be considered in quantizing the related side information. In general, correction vectors are harder to compress as they tend to be less correlated with each other than block-based depth values. Also, searching for the best correction vectors with a small search grid significantly increases the computational load as this vector is a two-dimensional quantity while depth is one-dimensional. We found that using a finer grid for depth search with a coarser correction vector resulted in similar RD performance.

5. Side information coding

This section describes the techniques used in encoding the side information for view synthesis. In Section 5.1, a lossless encoding of depths for MBs which choose the VSP as the best reference is described. It corresponds to the VSP 'with depth penalty' as defined in Section 3.1 and pertains to the coding scenario where coding efficiency improvement is the goal as will be demonstrated in Section 6.1. Next, in Section 5.2, lossy encoding of a depth map is discussed, which corresponds to the VSP 'without depth penalty' and is relevant to the coding scenario where the goal is offsetting the rate overhead as will be demonstrated in Section 6.2. Finally, Section 5.3 discusses a method applicable to both coding scenarios to improve side information coding efficiency.

5.1. Lossless encoding of depth using CABAC

With VSP, we encode a depth value and a correction vector for each MB that selects the VSP mode according to the RD mode decision as given by (7). The encoding of this information is done similarly to the CABAC encoding of motion vectors in H.264/AVC [1]. For instance, depth values are predicted and binarized in the same way as motion vectors, and similar context models are used. When an MB is chosen to use VSP, but has no surrounding MBs with the same reference, its depth is independently coded without any prediction. Similarly, each component of a correction vector is binarized using the fixed-length representation followed by CABAC encoding of the resulting bins. Note that correction vectors are not predictively encoded as they are usually not well correlated with their neighbors.

5.2. Lossy encoding of depth

We discussed in Section 5.1 the selective lossless encoding of depth information on a macroblock basis for view synthesis as determined by the RD-mode decision. However, there are application scenarios where a depth map for a frame needs to be compressed and sent to the decoder for rendering purposes such as intermediate view generation in FTV. In such cases, depth information for every region of a frame should be sent and will likely to incur a prohibitive bitrate overhead. This is particularly true if lossless compression is employed to code depth maps that are not temporally consistent. Therefore, lossy compression of depth maps is necessary. Conventional codecs such as H.264/AVC or MVC could be used for this purpose, as done in our experiments. Improved depth coding schemes that specifically account for depth attributes that are important to maximize intermediate view generation quality could also be considered, but this problem is beyond the scope of this paper. However, we show in Section 6.2 how the use of different quantization parameters (e.g., QPs in H.264/AVC) and sub-sampling ratios for depth compression impact the coding rate and the quality of synthesized prediction.

5.3. Improved disparity encoding with depth-to-disparity conversion

When encoding a disparity vector difference, we use disparity vectors from surrounding MBs for prediction. However, a neighboring MB with a VSP reference does not provide such information as it only has a depth/correction vector pair associated with it. The idea of depth-to-disparity conversion is to calculate the equivalent disparity vector corresponding to the depth/correction vector pair, then to use it as a prediction for disparity vector coding. Although not every pixel in a synthesized MB maps to a pixel in the reference frame separated by a single, identical disparity vector in general (e.g., non-translational inter-camera motion), we found it did not make any significant difference which pixel within the MB one chooses to find the corresponding disparity vector as it will be used only for predicting disparity vectors of surrounding MBs.

Given a depth value $D[c,t,x,y]$ that describes how far the object corresponding to pixel (x,y) is from camera c at time t, as well as the intrinsic matrix $A(c)$, the rotation matrix $R(c)$ and the translation vector $T(c)$, the pixel location $X_c = [x,y,1]^{\mathrm{T}}$ in the current camera frame c is projected into the world coordinate $[u,v,w]$ via (1). Next, the world coordinate is mapped into the target point $X_{c'}$ of the frame in camera c' which we wish to predict from via (2).

Then the direct transformation from the current camera to the target camera can be obtained by combining the two equations as follows:

$$X_{c'} = [x', y', z']^{\mathrm{T}} = M_1 \cdot D[c, t, x, y] + M_2, \quad (10)$$

where $M_1 = A(c') \cdot R^{-1}(c') \cdot R(c) \cdot A^{-1}(c) \cdot X_c$ and $M_2 = A(c') \cdot R^{-1}(c') \cdot \{T(c) - T(c')\}$. After normalizing $X_{c'} = [x', y', z']^{\mathrm{T}}$ by z', we get the point in the target camera $[x'/z', y'/z']$ corresponding to $[x, y]$ in the current camera and the disparity is derived as $\vec{D} = [x, y] - [x'/z', y'/z']$ where x', y', z' are given in (10).

6. Results and discussion

In this section, we show experimental results regarding the two multiview video coding scenarios as discussed in earlier sections. In Section 6.1, VSP is used to improve the coding efficiency of multiview video. In Section 6.2, the same technique is used to offset the rate overhead incurred by sending depth maps for intermediate view generation such as in FTV applications.

6.1. VSP for multiview coding efficiency improvement

The experimental results in this section are intended to demonstrate that the use of VSP in addition to DCP provides extra coding gains. Experiments were conducted using all the views (i.e., views 0–7, 100 frames per view) of the breakdancers and ballet sequences at 15 Hz. The sequences were encoded according to the MVC common conditions [12] such as the Hierarchical-B temporal prediction structure with a GOP size of 15 except that QPs of 27, 32, 37 and 40 were chosen instead of 22, 27, 32 and 37 as suggested in [12]. Our view synthesis techniques were built into the JMVM 1.0 software.

In Figs. 5 and 6, the curves labeled DCP correspond to the use of DCP and the ones labeled DCP+VSP represent the performance when using VSP in addition to DCP. All the standard intra-view (i.e., temporal and intra) predictions were also used in both cases. Note that the RD-optimal mode decision as described in Section 3.1 was made among all candidate types of reference pictures including VSP 'with depth penalty'. Hence, the bitrates in Figs. 5 and 6 include the rates for coding depth as an integral part of the multiview video bitstream.

Figs. 5(a) and 6(a) compare the RD performances with and without VSP averaged over all 100 frames of the B-views (i.e., views 1, 3 and 5), which are the views that utilize two spatially neighboring views from different directions in addition to temporal prediction. While the gains are not substantial at the higher bitrates, we do

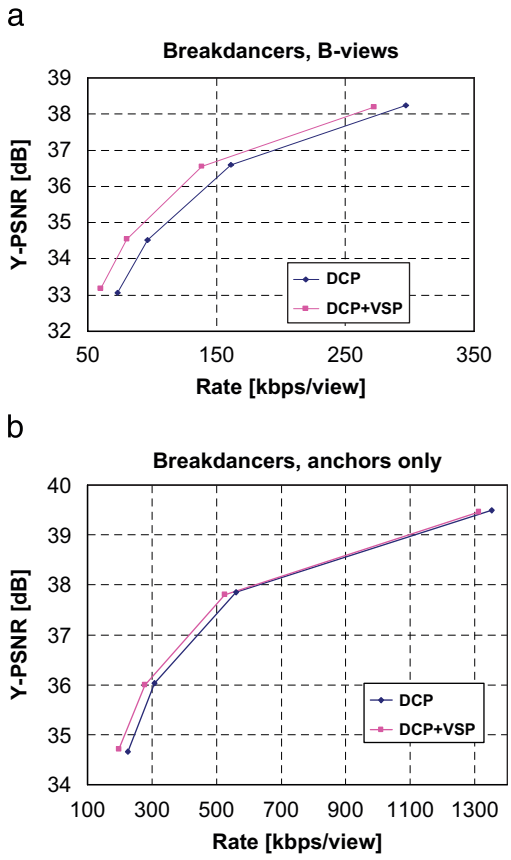

Fig. 5. RD comparison between DCP and DCP + VSP, Breakdancers. (a) Average over all B-views of breakdancers sequence (full GOP). (b) Average over anchor pictures in all views of breakdancers sequence.

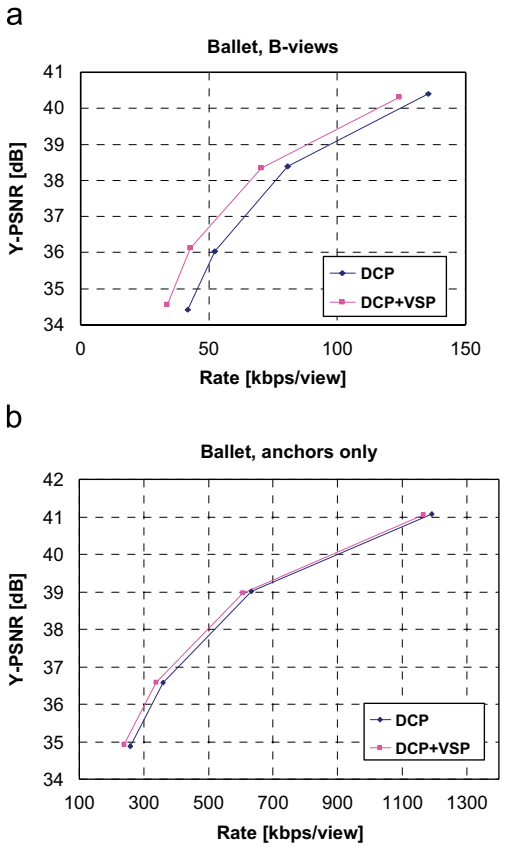

Fig. 6. RD comparison between DCP and DCP + VSP, Ballet. (a) Average over all B-views of ballet sequence (full GOP). (b) Average over anchor pictures in all views of ballet sequence.

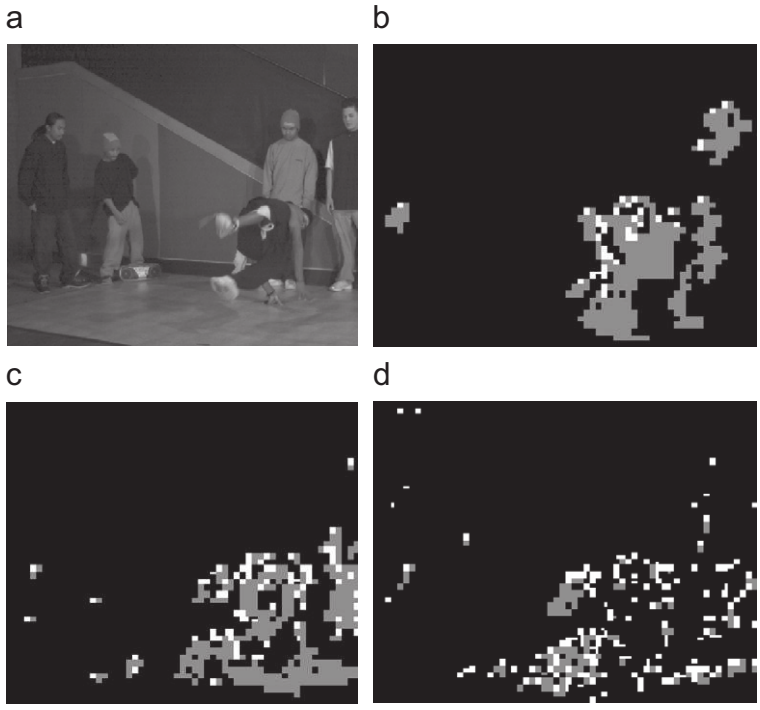

Fig. 7. RD-mode decision map for non-anchor picture of breakdancers. Black: non-VSP, gray: VSP (skip), white: VSP (non-skip). (a) View 1, frame 22. (b) QP = 40. (c) QP = 32. (d) QP = 22.

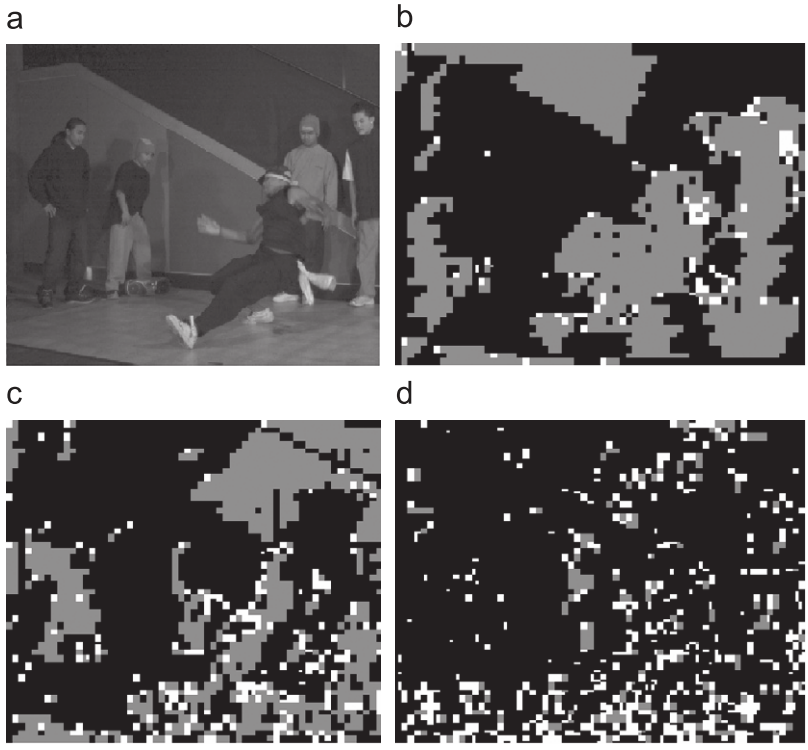

Fig. 8. RD-mode decision map for anchor picture of breakdancers. Black: non-VSP, gray: VSP (skip), white: VSP (non-skip). (a) View 1, frame 30. (b) QP = 40. (c) QP = 32. (d) QP = 22.

Table 1
Statistics of RD-mode decision for view 1 of breakdancers.

QP	Non-anchor (frame 22)		Anchor (frame 30)	
	% of VSP	% of S-skip	% of VSP	% of S-skip
40	11.2	10.1	42.1	39.5
32	13.8	11.3	36.0	29.6
22	7.6	3.4	15.6	4.9

observe notable gains in the middle to low bitrate range that are between 0.2 and 1.5 dB.

In Figs. 5(b) and 6(b) we examine the results averaged over all anchor pictures, which are pictures that do not employ temporal prediction and are used to facilitate random access points. These results include P-views, which are views that utilize one spatially neighboring view. It is shown that the use of VSP provides some small gains, but the average gains over all views are less than the gains observed in B-views. The main reason for this is that the use of view synthesis in addition to disparity compensation did not result in the same amount of relative bit savings for an anchor picture, which typically requires much higher bitrates to encode than non-anchor frames due to the lack of temporal prediction. This situation is aggravated for anchor pictures in P-views as they employ only one inter-view prediction.

Next, we analyze the performance of the RD mode decision. Figs. 7 and 8 show the resulting mode decision map for a non-anchor and anchor pictures of breakdancers with different QPs, respectively. In Table 1, we provide the percentages of 8×8 MBs using VSP or synthetic skip/direct modes (S-skip) in both cases. One can see that VSP is chosen more frequently in the anchor picture as no temporal references are being employed. Also, there is a tendency that more S-skip modes are chosen as QP becomes larger. The fact that many S-skip modes are chosen in the anchor picture case suggests that MBs through S-skip (using depth information) are often more useful as prediction than MBs through conventional skip/direct modes (using disparity vectors).

6.2. VSP for rate overhead reduction

In this section, we show experimental results on the use of proposed VSP in offsetting the rate overhead for sending depth maps in scenarios such as FTV. The first 16 frames of view 3 of breakdancers sequence as well as the corresponding depth maps provided by Microsoft were encoded according to the MVC common conditions [12] as in the previous subsection. The proposed technique was implemented in the same software used in the previous subsection with slight changes regarding the RD-mode decision for the VSP 'without depth penalty' as discussed in Section 3.1.

Figs. 9(a)–(d) show the results of encoding the depth map with QPs 22, 27, 32 and 37, respectively. View 3 of the multiview video (i.e., texture video) as well as the corresponding depth map were coded as B-views using the decoded views 2 and 4 for inter-view prediction. The vertical axis in each sub-figure is the (luma) PSNR of the encoded texture video, while the horizontal axis

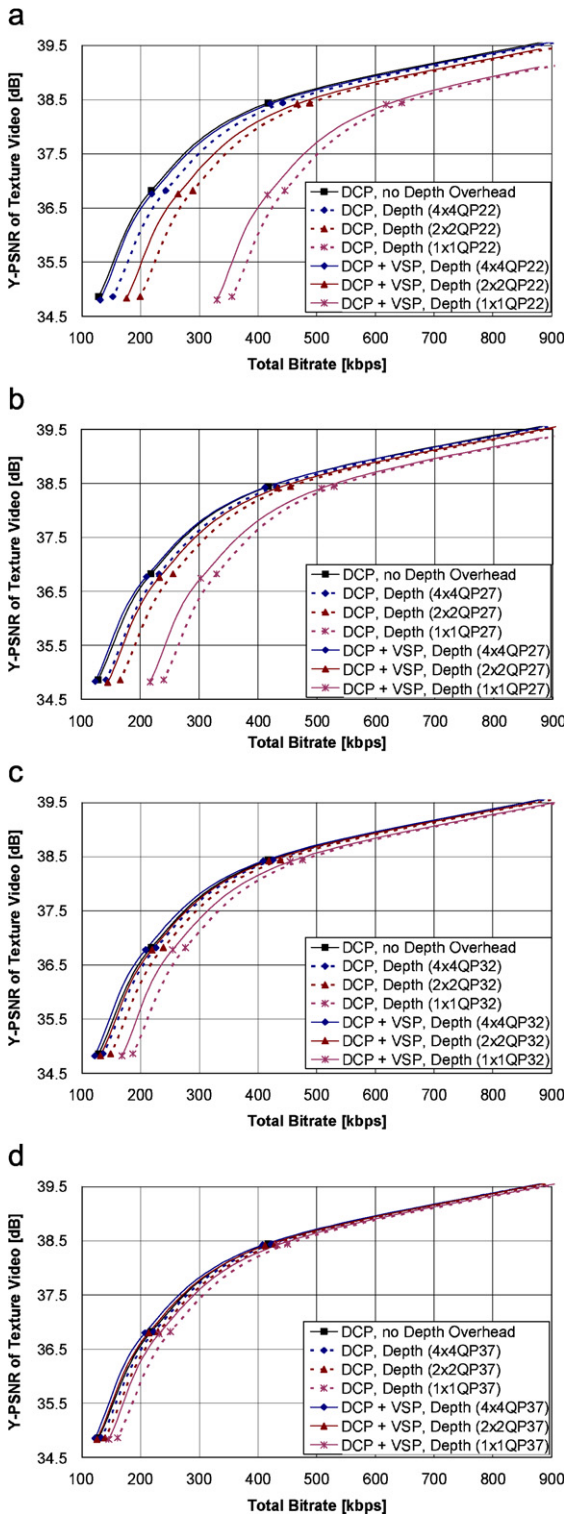

Fig. 9. Rate-overhead reduction via VSP (Breakdancers, view 3), 'total bitrate' = texture bitrate + depth bitrate. (a) QP for depth: 22. (b) QP for depth: 27. (c) QP for depth: 32. (d) QP for depth: 37.

corresponds to the sum of the bitrates used for encoding the texture and the depth maps.

The dotted curves correspond to the cases with different QPs and sub-sampling ratios (e.g., '4 × 4QP22' means the depth map is sub-sampled by four and encoded using QP of 22) for encoding depth maps. The solid curves with the same colors and markers correspond to the use of VSP for texture video coding using the encoded depth maps in addition to other types of prediction. As can be seen, the rate increase incurred by encoding depth maps is offset by view synthesis especially for large QPs and sub-sampling ratios. Fig. 10 illustrates this situation where the 'offsetting' is achieved by using less bits to code the multiview texture video by utilizing the depth information already available both at the encoder and the decoder for VSP as discussed in Section 3.1.

Fig. 11 shows a tendency that more synthesized prediction blocks are favored in the RD-decision as they are free of the depth coding penalty while often providing comparable prediction quality. It compares the number (in %) of 8 × 8 synthetic blocks (i.e., macroblocks chosen to use view-synthesized prediction) between the curves 'with depth penalty' vs. 'without depth penalty', which correspond to the re-encoding (based on the macroblock level RD-decision) vs. the re-use of the already available depth maps, respectively.

Note, however, that for small QPs or sub-sampling ratios, the rate overhead for coding depth maps increases significantly whereas the rate reduction via VSP thereof does not. For example, Figs. 12 and 13 show that the use of smaller sub-sampling ratio and QP for depth maps lead to somewhat limited improvement in the PSNR of the synthesized prediction [18], respectively. This implies that higher quality depth maps (as measured by PSNR) do not necessarily improve the quality of VSP significantly enough so that it could well offset the large rate overhead associated with higher-quality depth maps.

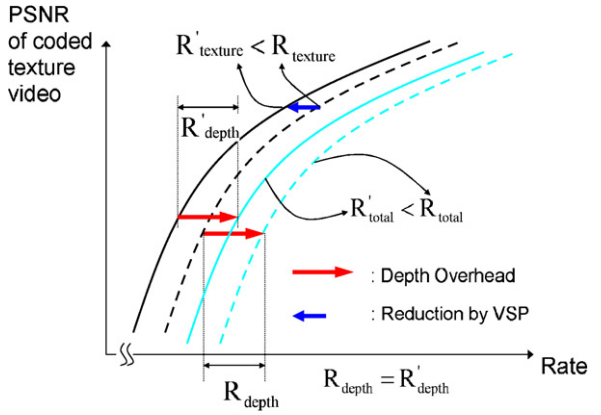

Fig. 10. Offsetting rate-overhead via VSP. Dotted lines: without VSP. Solid lines: with VSP.

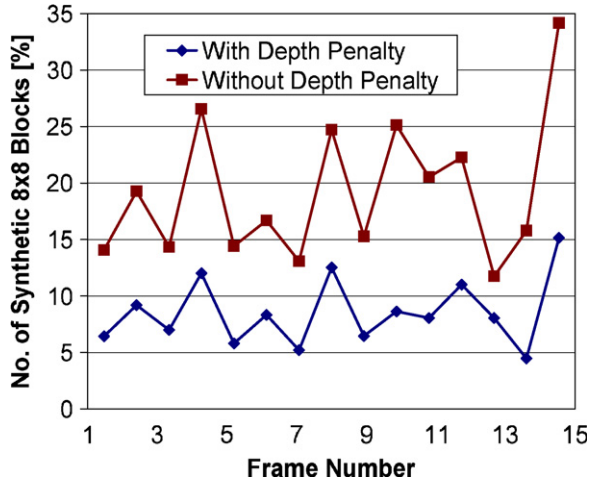

Fig. 11. Number of synthetic 8 × 8 blocks, view 3, QP = 27 for texture and depth. Average: 8.6% (with) and 19.2% (without).

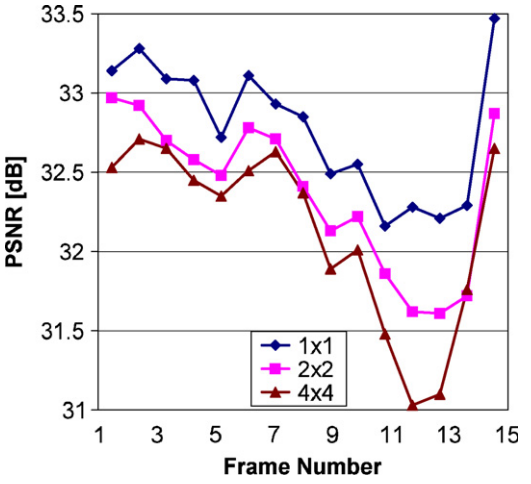

Fig. 12. Quality of synthesized prediction (avgerage PSNR = 32.8, 32.4 and 32.1 dB) when sub-sampling ratios for depth = 1, 2 and 4 with QP = 27 both for texture and depth.

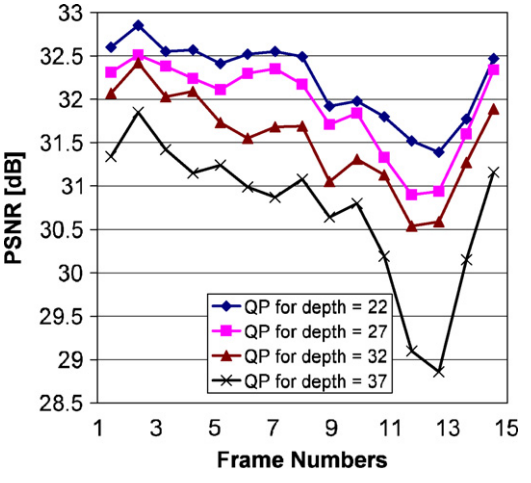

Fig. 13. Quality of synthesized prediction (avgerage PSNR = 32.2, 31.9, 31.5 and 30.7 dB) when QP for depth = 22, 27, 32 and 37 with sub-sampling ratio 4, QP = 32 for texture.

7. Concluding remarks

We proposed a rate-distortion optimized framework that incorporates view synthesis for improved prediction in multiview video coding. We described the means by which side information used for view synthesis prediction is generated and encoded. We also introduced a new synthetic skip/direct mode, which infers side information for view synthesis prediction from neighboring blocks. The proposed coding technique has shown to be effective at low to moderate bitrates, and especially for B-views that employ two spatially neighboring reference pictures. We also demonstrated that the rate overhead incurred by coding high-quality depth maps needed for rendering at the receiver in FTV applications can be offset by reducing the necessary bitrate for multiview (texture) video with the proposed technique. Some of the issues such as the effect of down-sampling as well as the use of different QPs for the depth map were also discussed.

There are many open issues that warrant future research. For one, we feel that an improved depth search algorithm would improve prediction efficiency. Also, the bit-allocation strategy considering inter-view dependency might allow for increased coding gains. Finally, prediction structures that utilize more bi-directional coding of views would seem to provide better overall results.

References

[1] Advanced video coding for generic audiovisual services, in: ITU-T Rec. H.264 & ISO/IEC 14496-10, 2005.
[2] N. Dodgson, Autostereoscopic 3D displays, IEEE Comput. 38 (8) (2005) 31–36.
[3] C. Fehn, 3D-TV using depth-image-based rendering (DIBR), in: Proceedings of Picture Coding Symposium (PCS), 2004, pp. 307–312.
[4] S. Ince, E. Martinian, S. Yea, A. Vetro, Depth estimation for view synthesis in multiview video coding, in: Proceedings of 3DTV Conference (3DTV-CON), Kos Island, Greece, 2007.
[5] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, Depth map creation and image based rendering for advanced 3DTV services providing interoperability and scalability, Signal Processing: Image Communication 22 (2) (2007) 217–234.
[6] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, C. Zhang, Multi-view imaging and 3DTV, IEEE Signal Processing Magazine 24 (6) (2007) 10–21.
[7] J. Lu, H. Cai, J. Lou, J. Li, An epipolar geometry-based fast disparity estimation algorithm for multiview image and video coding, IEEE Trans. Circuits Systems Video Technol. 17 (6) (2007) 737–750.
[8] E. Martinian, A. Behrens, J. Xin, A. Vetro, View synthesis for multiview video compression, in: Proceedings of the Picture Coding Symposium PCS, Beijing, China, 2006.
[9] E. Martinian, A. Behrens, J. Xin, A. Vetro, H. Sun, Extensions of H.264/AVC for multiview video compression, in: Proceedings of the IEEE International Conference on Image Processing, Atlanta, GA, 2006, pp. 2981–2984.
[10] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, Internat. J. Comput. Vision 47 (1–3) (2002) 7–42.
[11] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, Y. Yashima, View scalable multiview video coding using 3D warping with depth map, IEEE Trans. Circuits Systems Video Technol. 17 (11) (2007) 1485–1495.
[12] Y. Su, A. Vetro, A. Smolic, Common test conditions for multiview video coding, in: JVT-T207, Klagenfurt, Austria, 2006.
[13] M. Tanimoto, FTV (free viewpoint television) creating ray-based image engineering, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, Genoa, Italy, 2005, pp. 25–28.
[14] B. Wilburn, High performance imaging using large camera arrays, ACM Trans. Graph. 24 (3) (2005) 765–776.
[15] S. Yea, A. Vetro, RD-optimized view synthesis prediction for multiview video coding, in: Proceedings of the IEEE International Conference on Image Processing, vol. 1, San Antonio, TX, 2007, pp. I-209–I-212.
[16] S. Yea, A. Vetro, Report of CE6 on view synthesis prediction, in: JVT-W059, San Jose, CA, 2007.
[17] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, in: ACM Transactions on Graphics (Proceedings of the SIGGRAPH), vol. 23, Los Angeles, CA, 2004, pp. 600–608.
[18] L. Zhang, Fast stereo matching algorithm for intermediate view reconstruction of stereoscopic television images, IEEE Trans. Circuits Systems Video Technol. 16 (10) (2006) 1259–1270.