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

Analysis of View Synthesis Prediction Architectures in Modern Coding Standards

Tian, D.; Zou, F.; Lee, C.; Vetro, A.; Sun, H.

TR2013-076 September 2013

Abstract

Depth-based 3D formats are currently being developed as extensions to both AVC and HEVC standards. The availability of depth information facilitates the generation of intermediate views for advanced 3D applications and displays, and also enables more efficient coding of the multiview input data through view synthesis prediction techniques. This paper outlines several approaches that have been explored to realize view synthesis prediction in modern video coding standards such as AVC and HEVC. The benefits and drawbacks of various architectures are analyzed in terms of performance, complexity, and other design considerations. It is hence concluded that block-based VSP prediction for multiview video signals provides attractive coding gains with comparable complexity as traditional motion/disparity compensation.

SPIE Conference on Applications of Digital Image Processing XXXVI

Analysis of view synthesis prediction architectures in modern coding standards

Dong Tian[1], Feng Zou[1], Chris Lee[2], Anthony Vetro[1], Huifang Sun[1]
[1]Mitsubishi Electric Research Laboratories, 201 Broadway, 8th Floor, Cambridge, MA, USA
[2]National Cheng Kun University, 1 University Road, Tainan, Taiwan

ABSTRACT

Depth-based 3D formats are currently being developed as extensions to both AVC and HEVC standards. The availability of depth information facilitates the generation of intermediate views for advanced 3D applications and displays, and also enables more efficient coding of the multiview input data through view synthesis prediction techniques. This paper outlines several approaches that have been explored to realize view synthesis prediction in modern video coding standards such as AVC and HEVC. The benefits and drawbacks of various architectures are analyzed in terms of performance, complexity, and other design considerations. It is hence concluded that block-based VSP prediction for multiview video signals provides attractive coding gains with comparable complexity as traditional motion/disparity compensation.

Keywords: video coding, 3D, multiview, AVC, HEVC, view synthesis prediction

1. INTRODUCTION

Depth-based 3D formats enable the generation of virtual viewpoints, which can be used for advanced stereoscopic processing, free viewpoint video, and to generate the data necessary to drive multiview autostereoscopic displays **Error! Reference source not found.**. With the available depth images, intermediate views could be generated using Depth Image Based Rendering (DIBR) techniques with minimal complexity. The standardization of such formats is currently underway as extensions to both the H.264/MPEG-4 AVC and H.265/HEVC standards, and represents another step forward toward enhancing 3D functionality relative to existing multiview video formats [2][3]. With the depth information becoming an integral part of the data format, it is desirable to leverage this information for more efficient coding of the multiview input data. View synthesis prediction (VSP) is a key technique for joint texture-depth coding and the primary focus of this paper.

To be more precise, the data format used in the 3D extensions to H.264/MPEG-4 AVC and H.265/HEVC is composed of multiple texture view plus depth. That is, each texture view may be accompanied by its corresponding depth view, and such a format is often referred as multiview video plus depth (MVD) [4].

In general, view synthesis prediction is a technique that warps a picture from an adjacent viewpoint to the current viewpoint, and the warped picture serves as a reference picture to predict the current picture [5]. The warped picture is also known as a synthetic picture, which may be synthesized as a whole picture before encoding or decoding the current picture. Considering that only part of the synthetic picture is used for prediction, block based processing is preferred to reduce the run-time complexity and implementation burden. In this way, only the synthesized pixels that are used as predictors would be generated.

Depending on the viewpoint that the depth information is used to generate the synthetic picture, the VSP process can be realized with either a forward warping process or a backward warping process. With forward warping, the depth map from the reference viewpoint is used to project a pixel from the reference viewpoint to the current viewpoint, hence, the projected pixels may fall at sub-pixel positions in the current viewpoint due to the irregular projection procedure. Hence additional processing including hole filling and interpolation is needed to fill the pixel values at integer positions. Furthermore, forward warping is not friendly to block based processing as it is not straightforward to locate the reference area that is sufficient to produce a prediction block.

On the other hand, with backward warping, the depth map from the current viewpoint is used to fetch a pixel from the reference viewpoint. When the pixels being fetched are not at integer positions in reference viewpoint, they are rounded

to sub-pixel positions (e.g., quarter-pixel precision) and then interpolated using the same interpolation procedures as in regular motion compensation. With such a procedure, all integer-pixel positions will be filled with a value. In a typical backward view synthesis, hole pixels are detected by recording whether a pixel in the reference view is fetched more than one time, which requires extra processing. However, in backward warping, the extra process for hole detection is avoided, so there is not a significant burden in warping a pixel to several destinations. In such a way, the hole filling procedure can be avoided. While forward warping methods are considered to be slightly higher quality, the VSP schemes adopted in the 3D extensions are all based on backward warping due to its implementation advantages.

Since the depth map in the MVD data format is typically provided with pixel-level precision, each pixel may have a different disparity value. To achieve the maximum coding benefits from view synthesis prediction, each pixel would use an individual disparity for VSP; such a scheme is referred to as 1x1 VSP hereinafter (or 1x1 unconstrained VSP). In order to reduce the incurred complexity, block-based VSP has also been studied, where a block of pixels uses a single disparity vector converted from the corresponding depth block. Due to the loss of accuracy, the coding efficiency benefit of block-based VSP is less than that of 1x1 VSP.

In this paper, we perform an analysis of different VSP implementations with respect to performance, complexity and other design considerations. The remaining of the paper is organized as follows. Section 2 describes the specific VSP designs that have been considered during the development of the 3D extensions to the H.264/MPEG-4 AVC and H.265/HEVC standards. In section 3, we analyze the complexity of VSP and compare it with the traditional motion compensation method. In section 4, a performance evaluation of the different VSP configurations is conducted and summarized. Conclusions on VSP in the 3D extensions are provided in section 5.

2. VIEW SYNTHESIS PREDICTION IN MODERN CODING STANDARDS

View synthesis prediction has been recently adopted into the corresponding 3D extensions of both the H.264/MPEG-4 AVC and H.265/HEVC standards. Due to the particular designs of the base specifications, the VSP design for each standard has been adapted accordingly.

Generally, view synthesis prediction can be realized in two different ways. One way is via high-level modifications, that is, the synthetic picture is inserted to the reference picture lists, such that the synthetic picture is treated in the exactly same way as other reference pictures. This high-level approach maintains the existing architecture and would require minimal modification to the block-level syntax and processing. An alternative approach involves making lower level modifications to the syntax and decoding process, which is motivated by the fact that the synthetic picture is typically referenced with a zero motion vector. In the case of H.264/MPEG-4 AVC, the VSP predictor is often used in combination with the SKIP/DIRECT mode.

1.1 VSP in 3D-AVC

In the 3D extension to H.264/MPEG-4 AVC, which is referred to as 3D-AVC, view synthesis prediction is integrated with both high-level and low-level changes to the existing design in order to maximize the coding gains. In brief, VSP in 3D-AVC is realized and characterized by: 1) backward VSP prediction; and 2) 8x8 VSP, i.e., each 8x8 block uses a single disparity vector converted from an 8x8 depth block.

The initial VSP design in 3D-AVC used a separate VSP reference index in the reference picture lists to signal a synthetic reference [6]. However, it was later proposed to reuse the same reference picture index of an interview reference picture rather than adding a new VSP reference index [7]. This proposal adds a flag to differentiate the interview prediction and VSP prediction. With this high-level change, VSP is well-aligned and harmonized with traditional inter-prediction methods.

At the lower-level, it is important to efficiently signal a SKIP/DIRECT mode relative to VSP references. An additional flag has been introduced to the existing skip flag for this purpose [6][8]. A first flag signals whether a block is skipped or not. If it is skipped, the second flag further signal whether it refers to a synthetic reference or traditional reference.

The above design is based on the so-called “depth-first” coding order, where the depth component in a dependent view is coded prior to the texture component. However, there is also interest in supporting VSP with a texture-first coding order, as proposed in [9], such that the low-level design to support VSP in the 3D-HEVC framework, as described in the next section, would be imitated.

1.2 VSP in 3D-HEVC

In the 3D extension to HEVC, which is referred to as 3D-HEVC, view synthesis prediction is supported by low-level changes due to syntax mechanisms that are inherently supported by the HEVC design. Specifically, in HEVC, a merge candidate list was introduced to provide greater flexibility in the reference picture selection for a skipped block. In order to utilize this syntax efficiently, a special VSP candidate is appended to the merging candidate list [10]. When the VSP candidate is signaled, a synthetic reference block is then used as a predictor. Beside a single VSP candidate is inserted to the candidate list, more VSP candidates may be inherited from neighboring blocks coded in VSP mode during the candidate list construction.

Another major difference in 3D-HEVC compared to 3D-AVC is the coding order. In the current 3D-AVC draft specification, depth-first coding order is used; however, texture-first coding order has been selected in the 3D-HEVC design. The main consequence of the texture-first coding order is that the depth map from the current viewpoint will not be available when coding a texture view, which is required for backward view synthesis prediction. In order to overcome this problem, it was proposed to determine an approximated depth block for the purpose of backward view synthesis [11]. First, a disparity vector is derived from neighboring blocks if some of the blocks are coded in inter-view prediction modes or VSP modes; the process of deriving a disparity vector based on neighboring blocks is referred to as Neighboring Block Disparity Vector (NBDV). Then, using the derived disparity vector, a depth block is located in a reference view. Finally, the identified depth block is fetched and used as the depth block for the current texture block. In such a way, backward VSP can be performed. The 3-step procedure is illustrated in Figure 1. The VSP prediction in 3D-HEVC is currently limited to 4x4 blocks and the VSP is only performed for skip modes.

In 3D-HEVC, the concept of view synthesis prediction is not only applied in the pixel domain, but also to the motion field. For the motion field, the estimated depth block is used to derive a refined disparity vector derived from the neighboring blocks. In the current design, the corner pixels are used to produce a refined disparity vector and then the refined disparity is used to do a block based VSP. The technique is also known as depth-oriented NBDV (DoNBDV) [11][12].

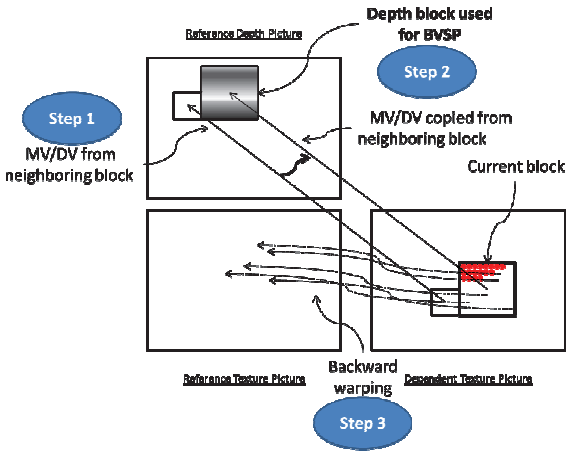

Figure 1. Illustration of backward VSP in 3D-HEVC.

1.3 VSP vs. Inter-view Prediction

Both view synthesis prediction and traditional inter-view prediction exploit inter-view similarities between multi-view videos. While disparity vectors are involved in both methods, view synthesis prediction is differentiated from traditional inter-view prediction with respect to the originality of disparity vectors.

With traditional inter-view prediction, the disparity vectors are estimated within an encoder purely for prediction purposes. Such disparity vectors are transmitted in the same way as regular motion vectors and the prediction/compensation procedure is exactly the same as temporal motion prediction/compensation. That is, the disparity vector is typically signaled for a prediction block. Signaling of such disparity vectors is an overhead.

However, for view synthesis prediction, the disparity vectors are converted from the accompanying depth maps. Such depth maps are provided as part of the data format for the purpose of rendering. Since depth is part of the data format, no overhead is assumed from the derived disparity vector. The downside of such disparity is that it is not optimal for predictive coding.

When having ideal depth map as inputs, view synthesis prediction is more likely to be selected when competing with traditional inter-view prediction. However, if the depth map contains errors or is optimized for subjective rendering quality, then the selection between view synthesis prediction and traditional inter-view prediction will be compromised between the prediction accuracy of depth/disparity and the overhead of transmitting an optimized disparity vector.

1.4 VSP precisions and Constrained VSP

From the above comparison between VSP and inter-view prediction, there is interest in aligning the implementation of VSP with traditional motion compensation (MC) methods. Such a harmonization may reduce hardware implementation costs, but the performance would be impacted as well.

With a conventional VSP method (1x1 unconstrained VSP), the disparity vector is converted from the depth map and each pixel may have an individual disparity vector, as illustrated in Figure 2. On the other hand, conventional MC is performed on a block basis, e.g. the prediction unit may be an MxN block. If a single disparity vector is converted to an MxN block, the VSP is referred to as MxN VSP hereinafter. In such a way, the conventional MC module need not be changed and it can carry out MxN VSP prediction. Given such advantages, block based VSP has been adopted to both the 3D-AVC and 3D-HEVC designs. MxN is equal to 8x8 in 3D-AVC and 4x4 in 3D-HEVC.

In order to take advantage of the coding benefits from using an accurate disparity vector per pixel, a constrained 1x1 VSP has been proposed [13] to reduce the off-chip data transfer rates, which is a major bottleneck of 1x1 unconstrained 1x1 VSP. On the left of Figure 3, the pixels that need to be fetched for traditional compensation are included in the dashed rectangle of size (M+7)x(N+7), while on the right of Figure 3, the pixels that need be fetched for a constrained 1x1 VSP is a rectangle of size (M+7+α)xN. If the window size from the traditional compensation is set as an upper limit, we can solve for the α values that satisfy (M+7+α) <= (M+7)x(N+7) given each block size MxN. Table 3 shows the derived α values and corresponding horizontal size M+α. For chroma components, similar α values can be derived. The off-chip data transfer rate will be studied in the next section, where the 1x1 CVSP is compared with other VSP methods.

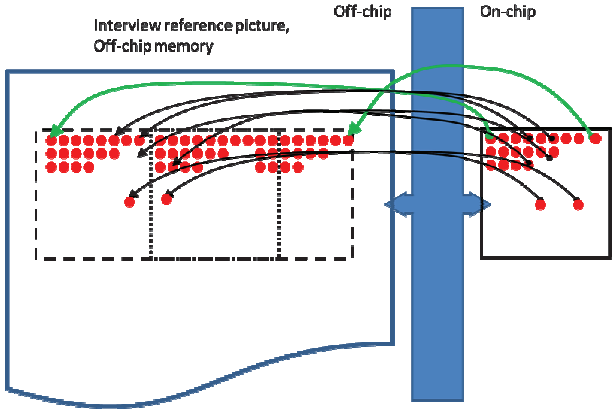

Figure 2. Illustration of unconstrained 1x1 VSP prediction.

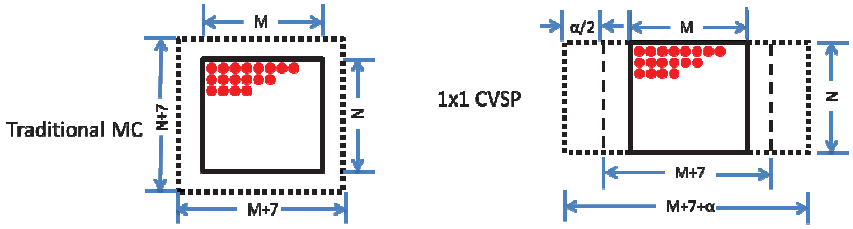

Figure 3. Illustration of traditional MC vs. a constrained 1x1 VSP prediction.

Table 1. Summary of off-chip data transfer rates.

M	N	α	M+α (size)	-α/2 (starting)
4	8	10	14	-5
8	4	26	34	-13
8	8	13	21	-6
8	16	7	15	-3
16	8	20	36	-10
16	16	10	26	-5
16	32	5	21	-2
32	16	17	49	-8
32	32	9	41	-4
32	64	4	36	-2
64	32	16	80	-8
64	64	8	72	-4

3. COMPLEXITY ANALYSIS ON VIEW SYNTHESIS PREDICTION

In terms of software implementation, VSP compensation does not incur a significant increase in run-time, only about 3% decoding time increase [13]. However, the impact on hardware requires more through study and analysis. In this section, the hardware complexity is analyzed relative to traditional motion compensation (MC) with different VSP compensation (VSP MC) methods.

Several assumptions are made herein that reflect typical implementation practices for hardware.

- The reference picture buffer is stored in an off-chip memory, referred to as DPB buffer;
- Quarter-pixel accuracy is applied for traditional block-based motion compensation with separable horizontal and vertical interpolation filters; 8-tap for half pixel position and 7-tap for quarter-pixel position in HEVC [14][15]. The same quarter-pixel accuracy is applied for VSP compensation as well;
- Prediction block (PB) size is represented as MxN, where M and N denote the width and height of the PB, respectively;
- The interpolation process is enforced on an on-chip Processing Elements (PE), using the on-chip memory with data fetched from the off-chip DPB buffer.

There are many factors that need be evaluated with respect to hardware implementation complexity. For example, the off-chip data transfer rate, pattern, on-chip memory access and the number of interpolations. The following four compensation methods are compared in terms of complexity: Traditional MC, 4x4 VSP MC, 1x1 VSP MC, and 1x1 CVSP MC.

1.5 Off-chip memory architecture

The off-chip memory used for reference picture buffers is normally realized with DRAMs (dynamic random-access memory). In a typical AVC or HEVC embedded decoder, a reconstructed picture is written to the off-chip memory on a block basis, if it is used as a reference. Such a writing procedure is the same for all the different MC methods.

At the time when the pixels of the reconstructed picture need to be fetched, a random access to the DRAM is normally required for traditional MC due to the non-zero motion vectors. The starting address in the reference picture buffer can be formularized as:

$$AddrRef = AddrCurr + MVy * Stride + MVx \quad (1)$$

where AddrCurr represents the corresponding address of the collocated block in the reference picture. Stride represents the memory size occupied by a line of pixels, which may include the padded pixels. MVx and MVy are horizontal and vertical components of motion vectors.

Picture storage architecture is a significant factor for efficient data reading. A first question is whether the design of the architecture needs to be changed to accommodate any of the VSP MC methods. We consider 1x1 CVSP as an example in the following to show that the memory design does not need to be modified while still maintaining the efficiency.

Following the description in Section 2.4, the starting address in the reference picture buffer for 1x1 CVSP can be expressed as:

$$AddrRef = AddrCurr + DVx - \alpha/2 \quad (2)$$

Compared to (1), both have two random variables. In (1), the random variables are MVx and MVy, while in (2), they are DVx and -α/2. In theory, the distribution of DVx is identical to MVx. On the other hand, –α/2 falls between -2 and -13, which is a smaller range than the typical range for MVy. Hence, it can be concluded that the off-chip DRAM architecture does not need to be modified to incorporate the VSP MC methods, as all the MC methods are a random access to the off-chip memory.

In the following, we investigate the frequency of fetching pixels from off-chip memory, which impacts the run-time complexity. Line buffer access is assumed such that a line of pixels would be fetched in a single request.

For traditional block-based MC, the number of fetches is equal to the number of lines in an MxN block, that is, N+7 when considering 7 additional lines are needed to perform interpolation.

For 4x4 VSP MC, the block of MxN is split into MxN/(4x4) sub-blocks of size 4x4. Since no vertical interpolation is required, each 4x4 requires 4 lines to be fetched. In total, the number of fetches is Nx(M/4) for an MxN block.

For 1x1 VSP MC, no line buffer access can be utilized for optimizations since a random distribution of DVx from pixel to pixel is assumed. Thus, each pixel would incur one data fetch. For an MxN block, the number of fetches is then MxN.

For 1x1 CVSP MC, a block of MxN would just require N lines to be fetched since there is no vertical interpolation.

As an example, consider a 64x64 block with 8x4 sub-partitioning. The total number of fetches is summarized in Table 2. From the table, it can be found that 4x4 VSP requires fewer fetches, while 1x1 unconstrained VSP needs about 4x more fetches which would be a major burden for implementation. However, the number of data fetches for the 1x1 CVSP is significantly reduced to about 40% of the traditional MC.

Table 2. Summary of off-chip data fetching numbers.

	Traditional MC	4x4 VSP	1x1 VSP	1x1 CVSP
Off-chip data fetching time	11*128 = 1,408	8*128 = 1,024	32*128=4,096	4*128=512

1.6 Off-chip data transfer rate

Off-chip data transfer rate is another important factor to evaluate the algorithm complexity. The data rate impacts both the architecture design and run-time complexity. In the following analysis, we assume the same 8-tap interpolation is applied on all MC methods. The maximum possible number of pixels is counted for the worst case condition.

As noted in the previous section, traditional MC requires 7 additional neighboring integer pixels along both the horizontal and the vertical boundaries of the prediction block. Therefore, the number of pixels being fetched from off-chip DPB memory is (M+7)x(N+7). The actual number may be higher due to different architecture designs, but it is fine to apply the same assumption for other MC methods as well.

For 4x4 VSP MC, the prediction block of an MxN is split into MxN/(4x4) sub-blocks with 4x4 sizes. Each 4x4 block needs to fetch 4x(4+7) pixels from DPB, considering that there is no vertical interpolation in VSP prediction. Therefore, the number of pixels being fetched is 4x(4+7)xMxN/(4x4).

For 1x1 unconstrained VSP MC, each pixel needs to fetch 8 pixels from DPB in the horizontal direction, so the total number is MxNx8.

For 1x1 constrained VSP MC, the pixels that need to be fetched is (M+7+α)xN.

Considering again the example of a 64x64 block with 8x4 sub-partitioning, (i.e., MxN = 8x4), there are 128 8x4 partitions. The total off-chip data transfer rates required for different MC methods are summarized in Table 3. It is evident that 4x4 VSP MC has a 50% lower data rate compared to traditional MC, while 1x1 unconstrained VSP MC incurs a 50% increase. Therefore, 4x4 VSP MC is feasible for a practical implementation, while 1x1 unconstrained VSP would is more expensive to realize in hardware. However, the 1x1 CVSP substantially reduces the required data rate, and is even slightly less than traditional MC and is hence feasible to be implemented.

Table 3. Summary of off-chip data transfer rates.

	Traditional MC	4x4 VSP	1x1 VSP	1x1 CVSP
Off-chip data transfer rate	165*128 = 21,120	88*128 = 11,264	256*128=32,768	164*128=20,992

1.7 On-chip data access

For on-chip memory access, the complexity of 1x1 VSP is higher than traditional MC or 4x4 VSP. The source of the increased complexity is due to the fact that a unique vector can be used in traditional MC or 4x4 VSP, while 1x1 VSP cannot guarantee such a regular access pattern. A regular access pattern may lead to data reuse for data loading.

Another benefit of regular access is that the interpolation coefficients only need to be loaded once. In case of 1x1 VSP, the interpolation filter may be switched between 8-tap filter and 7-tap filter, which may require loading different filter coefficients from pixel to pixel.

However, in order to interpolate one sub-pixel, the 4x4 or 1x1 VSP compensation needs to load 8 (or 7) integer pixels from on-chip memory; while traditional MC needs to load 8x8=64 integer pixels in the worst case.

1.8 Number of interpolations

The last factor to be evaluated is the number of interpolations required for each compensation method. The required number of arithmetic operations is proportional to the number of interpolations.

For traditional MC, the worst case happens for the sub pixel located at j_{00} as shown in Figure. 3. If data reuse is not considered, (8+1)xMxN interpolations are incurred. If vertically interpolated data can be reused, the number of interpolation can be reduced to (N+8)xM+NxM = (2N+8)xM, but it may lead to a more complex circuit design in order to save the computation power.

For 4x4 VSP MC, the prediction block MxN is split into MxN/(4x4) sub-blocks of size 4x4. Each 4x4 need to apply 4x4 = 16 interpolations, as there is no vertical interpolation. Therefore, 4x4xMxN/(4x4) = MxN interpolations are incurred.

For 1x1 VSP MC, the total number of interpolation is MxN.

In the example of a 64x64 block with all 8x4 sub-partitions, the number of interpolations is summarized in Table 4. It can be found that the number of interpolations of all VSP methods only require 25% computation amount compared to traditional MC if data reuse is applied for traditional MC.

Table 4. Summary of number of interpolations.

	Traditional MC	4x4 VSP	1x1 VSP	1x1 CVSP
Off-chip data fetching time	288*128 = 36,864 (w/o data reuse) 128*128 = 16,384 (w/data reuse)	32*128 = 4,096	32*128 = 4,096	32*128 = 4,096

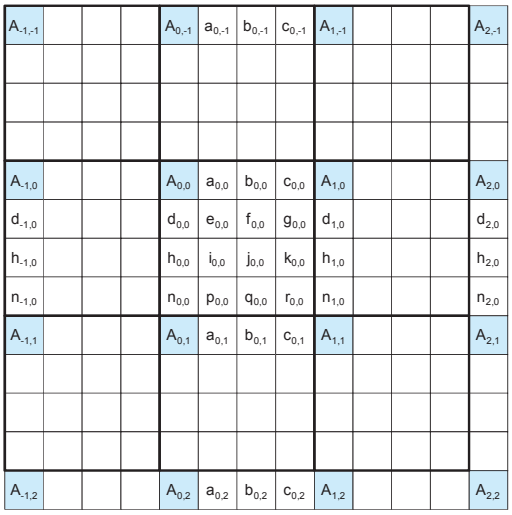

Figure 3. Integer and fractional positions for HEVC-based 8/7 tap interpolation filter.

1.9 Overall comparison of complexity

From the analysis presented in the previous sections, the following summary regarding the impact and feasibility of different VSP configurations on decoder complexity can be made:

- Block based VSP, e.g. 4x4 VSP, is a feasible design and there is no a bottleneck to realize this configuration with a hardware platform.
- Unconstrained 1x1 VSP incurs much higher complexity than traditional MC, especially with regards to the off-chip memory access; as a result, this configuration is not considered to be a feasible candidate for practical implementations.
- Constrained 1x1 VSP results in comparable off-chip memory complexity relative to traditional MC with similar or higher on-chip complexity. While the arithmetic complexity of 1x1 CVSP is much simpler, this configuration does require modifications to existing MC modules. We believe 1x1 CVSP is feasible to implement, but these modifications need to be justified by sufficient coding benefits, which are studied further in the next section.

Although the focus of our complexity analysis was on the decoder, the same analysis applies for the encoder as well. Additionally, all VSP predictions do not require any motion estimation at the encoder. Hence, the run-time complexity introduced by VSP method at the encoder is considered minimal.

4. PERFORMANCE EVALUATION

In this section, the performance of VSP methods is evaluated in 3D-AVC and 3D-HEVC. Four HD sequences are tested. Two of them are natural scenes, Poznan_Hall (indoor) and Poznan_Street (outdoor), which are provided by Poznan University; the other two are computer graphics (CG) sequences, Undo_Dancer (static camera) and GT_Fly (moving camera), which are provided by Nokia Research Center. The depth maps for the natural sequences were estimated using stereo matching algorithms, and those for the CG sequences are generated from 3D models.

Three pairs of texture and depth video are coded. The inter-view prediction pattern is "PIP", where the center view is coded as base view and the two side views are coded as dependent views. Two intermediate views are to be synthesized from the three coded views and used to evaluate the coding performance. Three overall Bjøntegaard bitrate savings are calculated: video PSNR vs. video bitrate, video PSNR vs. total bitrate and synthesis PSNR vs. total bitrate. Note that the

synthesis PSNR is calculated by referencing the synthesis picture generated from uncompressed texture and depth views. The common test conditions given in [16] are followed.

Firstly, we evaluate the performance of the VSP methods currently adopted within the draft standards. For 3D-AVC, 8x8 VSP was adopted, i.e., a single disparity vector is derived for an 8x8 block. Due to the recent software was not available when preparing the paper, ATM 7.1 is used for 3D-AVC evaluation [17], where 4x4 VSP is in use. However, the coding performance is quite similar to 8x8 VSP [7]. Table 5 summarizes the experimental results and shows ~3% bitrate savings. For 3D-HEVC, 4x4 VSP was adopted. HTM 6.0 is used for 3D-HEVC evaluation **Error! Reference source not found.** . Table 6 summarizes the experimental results and shows 1.3% and 1.1% bitrate savings for coded and synthesis results, respectively.

In addition, the 1x1 CVSP was also implemented within HTM 6.0 software and evaluated in context of 3D-HEVC. As shown in Table 7, the bitrate savings are 1.6% and 1.4% for coded and synthesis results, respectively. When compared to the adopted 4x4 VSP, the additional gain seems quite marginal, which does not justify the use of 1x1 CVSP at this time. Further work is underway to improve the performance of 1x1 CVSP.

Table 5: Performance comparison of 3D-AVC with 4x4 VSP on vs 4x4 VSP off.

	video 0	video 1	video 2	video PSNR/ video bitrate	video PSNR/ total bitrate	synth PSNR/ total bitrate
Poznan_Hall2	0.0%	-3.1%	-4.2%	-1.2%	-1.1%	-1.2%
Poznan_Street	0.0%	-13.7%	-12.1%	-3.5%	-3.2%	-2.9%
GT_Fly	0.0%	-27.0%	-26.1%	-6.3%	-5.8%	-5.3%
Undo_Dancer	0.0%	-11.8%	-10.7%	-3.0%	-2.7%	-2.7%
Average	0.0%	-13.9%	-13.3%	-3.5%	-3.2%	-3.0%

Table 6: Performance comparison of 3D-HEVC with 4x4 VSP on vs 4x4 VSP off.

	video 0	video 1	video 2	video PSNR/ video bitrate	video PSNR/ total bitrate	synth PSNR/ total bitrate
Poznan_Hall2	0.0%	-0.3%	-1.5%	-0.3%	-0.2%	-0.4%
Poznan_Street	0.0%	-2.9%	-2.7%	-0.9%	-0.8%	-0.7%
GT_Fly	0.0%	-8.6%	-8.7%	-2.3%	-2.2%	-1.7%
Undo_Dancer	0.0%	-8.4%	-7.5%	-2.3%	-2.1%	-1.5%
Average	0.0%	-5.1%	-5.1%	-1.4%	-1.3%	-1.1%

Table 7: Performance comparison of 3D-HEVC with 1x1 CVSP on vs 1x1 CVSP off

	video 0	video 1	video 2	video PSNR/ video bitrate	video PSNR/ total bitrate	synth PSNR/ total bitrate
Poznan_Hall2	0.0%	-0.9%	-1.7%	-0.5%	-0.4%	-0.5%
Poznan_Street	0.0%	-3.0%	-2.8%	-0.9%	-0.8%	-0.7%
GT_Fly	0.0%	-8.9%	-9.1%	-2.4%	-2.3%	-1.8%
Undo_Dancer	0.0%	-11.8%	-10.3%	-3.3%	-3.0%	-2.5%
Average	0.0%	-6.2%	-6.0%	-1.8%	-1.6%	-1.4%

5. CONCLUDING REMARKS

This paper presents several architectures to enable view synthesis prediction within modern coding standards such as H.264/MPEG-4 AVC and H.265/HEVC. The general concept of view synthesis prediction and its impact on different architectures including forward VSP vs. backward VSP, pixel based VSP vs. block based VSP and unconstrained VSP vs. constrained VSP have been reviewed. A performance evaluation in terms of coded data and synthesized views has also been provided.

From the experiments and analysis that have been conducted, it is shown that enabling of block-based VSP prediction for multiview video signals provides an attractive coding gain with comparable complexity as traditional motion/disparity compensation. Moreover, a constrained pixel-based VSP is still under study. Although the additional coding benefits have not fully been justified with the current test materials and evaluation criteria, we believe that pixel-based VSP is also feasible in terms of hardware implementation, and that further coding efficiency benefits could be realized, especially when considering subjective quality assessment techniques, which is a topic for further study.

REFERENCES

[1] Urey H., Chellephan K. V., Erden E., and Surman P., "State of the Art in Stereoscopic and Autostereoscopic Displays," *Proceedings of the IEEE*, vol. 99, no. 4, 540-555 (2011).

[2] Vetro A., Wiegand T., and Sullivan G. J., "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/AVC Standard", *Proc. of the IEEE*, vol. 99, no. 4, 626 - 642 (2011).

[3] Müller K., Merkle P., and Wiegand T., "3D Video Representation Using Depth Maps", *Proc. of the IEEE*, vol. 99, no. 4, 643 - 656 (2011).

[4] Video Group, "Report on Experimental Framework for 3D Video Coding," ISO/IEC JTC1/SC29/WG11 MPEG Doc. N11631, Guangzhou, CN (2010).

[5] Yea S., and Vetro A., "View Synthesis Prediction for Multiview Video Coding", *Signal Processing: Image Communication*, vol. 24, no. 1+2, pp. 89-100 (2009).

[6] Tian D., Graziosi D., Wang Y. Cheung N.-M., Vetro A., "Mitsubishi Response to MPEG Call for Proposal on 3D Video Coding Technology", ISO/IEC JTC1/SC29/WG11 MPEG Doc. m22663, Geneva, CH (2011)

[7] Su W., Rusanovskyy D., Hannuksela M., "3D-CE1.a related: Harmonization of inter-view and view synthesis prediction in 3D-AVC", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-D0162, Incheon, KR (2013).

[8] Lee J. Y., Lee J., and Park D.-S., "3D-AVC-CE1 results on Samsung's inter-view prediction using view synthesis", ISO/IEC JTC1/SC29/WG11 MPEG Doc. m23658, San Jose, US (2012)

[9] Chen Y., Kang J., Zhang L., Zhao X., Wang Y.-K., Joshi R., Karczewicz M., "CE2.a related: MB-level NBDV for 3D-AVC", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-D0185, Incheon, KR (2013).

[10] Zou F., Tian D., Vetro A., Shimizu S. Sugimoto S. Kimata H., "CE1.h: Results on View Synthesis Prediction", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-B0102, Shanghai, CN (2012).

[11] Tian D., Zou F., Vetro A., "CE1.h: Backward View Synthesis Prediction using Neighbouring Blocks", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-C0152, Geneva, CH (2013).

[12] Chang Y.-L., Wu C.-L., Tsai Y.-P., Lei S., "3D-CE1.h: Depth-oriented Neighbouring Block Disparity Vector (DoNBDV) with Virtual Depth Retrieval", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-C0131, Geneva, CH (2013).

[13] Zou F., Tian D., Vetro A., Sun H., "3D-CE1.h: VSP Complexity Analysis and Constrained VSP (CVSP)", Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-D0165, Incheon, KR (2013).

[14] High Efficiency Video Coding, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.

[15] Sullivan G. J., Ohm J.-R., Han W.-J., and Wiegand T., "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649–1668, Dec. 2012.

[16] Rusanovskyy D., Müller K., Vetro A., "Common Test Conditions of 3DV Core Experiments," Joint Collaborative Team on 3D Video Coding Extension Development, Doc. JCT3V-D1100, Incheon, KR (2013).

[17] SVN repository for ATM 7.1, http://mpeg3dv.research.nokia.com/svn/mpeg3dv/tags/3DV-ATMv7.1

[18] SVN repository for HTM 6.0, https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-6.0/