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

View Synthesis Prediction Using Skip and Merge Candidates for HEVC-based 3D Video Coding

Zou, F.; Tian, D.; Vetro, A.

TR2013-032 May 2013

Abstract

Traditional multi-view coding (MVC) systems compress the texture content captured from different view points, where temporal and inter-view redundancy are exploited to improve MVC coding efficiency. The advanced 3D video coding systems compress both the texture content and its corresponding depth captured from different view points, known as multiview video plus depth (MVD), to support low complexity free view point applications. However, MVD systems consist of a large amount of data including both texture and depth to be compressed and transmitted. To improve the coding efficiency of MVD systems, view synthesis prediction (VSP) can be used to further reduce inter-view redundancy using synthetic views as predictors. In this paper, an in-loop view synthesis framework is proposed, where the synthesized predictor is encoded as a special motion compensated predictor and the motion information is encoded as one of the motion predictors in skip/merge candidate list for HEVC-based 3D video coding. The proposed scheme is applicable to both texture coding and depth coding. The experimental results show that the proposed framework improved the coding performance up to 12.1% for dependent views.

IEEE International Symposium on Circuits and Systems (ISCAS)

View Synthesis Prediction Using Skip and Merge Candidates for HEVC-based 3D Video Coding

Feng Zou, Dong Tian, Anthony Vetro
Mitsubishi Electric Research Laboratories
201 Broadway, 8th Floor
Cambridge, MA 02139, USA
Email: {fzou,tian,avetro}@merl.com

***Abstract*— Traditional multi-view coding (MVC) systems compress the texture content captured from different view points, where temporal and inter-view redundancy are exploited to improve MVC coding efficiency. The advanced 3D video coding systems compress both the texture content and its corresponding depth captured from different view points, known as multi-view video plus depth (MVD), to support low complexity free view point applications. However, MVD systems consist of a large amount of data including both texture and depth to be compressed and transmitted. To improve the coding efficiency of MVD systems, view synthesis prediction (VSP) can be used to further reduce inter-view redundancy using synthetic views as predictors. In this paper, an in-loop view synthesis framework is proposed, where the synthesized predictor is encoded as a special motion compensated predictor and the motion information is encoded as one of the motion predictors in skip/merge candidate list for HEVC-based 3D video coding. The proposed scheme is applicable to both texture coding and depth coding. The experimental results show that the proposed framework improved the coding performance up to 12.1% for dependent views.**

I. Introduction

In recent years, 3D video content has become more and more prevalent in both the movie industry and home entertainment applications. At the same time, the manufacturing cost of 3D displays has been reduced due to the developed 3D display technologies, which undoubtedly promotes the spread of 3D video content as a result. However, due to the large data size of 3D video content, it is still an open issue on how to efficiently store and transmit 3D video content. Therefore, it is desirable to establish a 3D video coding standard to incorporate the state-of-the-art coding techniques to improve the coding efficiency, which would not only enable the interoperability of 3D video streams, but prosper the 3D video industry as well. Based on this demand, the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T WP3/16 and ISO/IEC JTC 1/ SC 29/ WG 11 was established in July 2012. The primary goals of the JCT-3V are to specify 3D video coding extensions of the Advanced Video Coding (AVC) and the High Efficiency Video Coding (HEVC) standards.

The multiview video coding (MVC) extension of AVC supports efficient compression video content of a scene captured from different view points using synchronized video cameras. In MVC, besides the temporal redundancy, there is significant redundancy existing between the different viewpoints of the texture component. Temporal motion compensated prediction (MCP) via block matching is used to reduce the temporal redundancy represented by a motion vector; while disparity compensated prediction (DCP) via block matching is used to reduce the inter-view redundancy wherein samples from one viewpoint are predicted from samples in a different viewpoint represented by a disparity vector [1]. When the input data is composed of the texture component only in MVC, it requires high computation complexity to estimate the depth map, which is undesirable in a practical free view point application scenario.

To achieve the goal of a low complexity free view point system, the MVD data format has been selected as the basis for 3D standards that facilitate intermediate view generation. In MVD, although adding the depth information into the input data undoubtedly imposes a heavier burden on the 3D video coding, the depth map can be utilized to provide better prediction of the texture component, as done in our previous work in [2], known as View Synthesis Prediction (VSP).

The basic idea of VSP is to generate a block predictor for the target block by warping pixel-by-pixel values using the reference view texture and depth. In [2], one synthesized virtual view was added in the reference list for non-translational disparity compensated prediction before encoding the current view. Based on [2], a rate-distortion optimized VSP was proposed by incorporating the block-based depth and correction vectors in [3]. Using VSP, [4] proposed to generate a scalable enhancement view predictor, where the base views and the residue of enhancement views are encoded by the conventional video coding. In [5], a general VSP is developed by extending the warping source from one view to two views as well as applying VSP to both texture and depth components.

Although the techniques mentioned above improve the coding efficiency of MVD coding, they fail to fully utilize the existing coding modes to achieve a unified coding design with high coding efficiency. In this paper, to further efficiently represent the VSP mode, a VSP candidate is included and signaled in the skip and merge candidate list rather than simply adding a synthetic frame in the reference list. The proposed scheme can be regarded as a generalization of HEVC standard with depth-assisted prediction. At the encoder, the VSP candidate is evaluated against other traditional spatial and temporal motion predictors according to the rate-distortion criteria. To further efficiently represent the VSP candidate, a pruning process is applied to eliminate duplicate candidates in order to reduce the overhead needed to represent the candidates in the list.

The rest of this paper is organized as follows. Section II provides a brief review of the coding structure used in JCT-3V and demonstrates the frame level VSP generation. In Section III, the process of VSP is elaborated in detail. In Section IV, the skip and merge candidate lists in HEVC-based 3D coding are investigated first. Subsequently, extending the skip and merge candidate lists is proposed by incorporating the VSP candidate as a motion predictor candidate. In Section V, extensive simulations are conducted to evaluate the performance of skip/merge candidate list based VSP representation. Finally, conclusions are presented in Section VI.

II. Coding Structure

In our work, we assume that a hierarchical B coding structure is used to exploit the temporal redundancy while IPP coding structure is used to exploit the inter-view redundancy as shown in Fig.1. At each time instance, the base view is firstly encoded followed by two dependent views. For each view, the base view can only refer to the

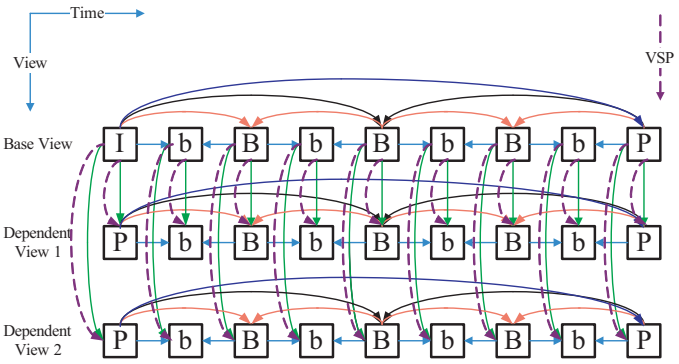

Fig. 1. Sample Coding Structure for the three view case

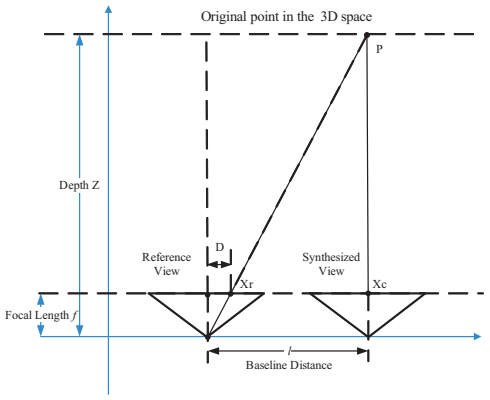

Fig. 2. Depth-assisted image rendering

previously coded base views as reference frames. The dependent view can refer to both the previously coded base view and its previously coded temporal views as reference views. For each view at each time instance, the texture component is coded prior to the depth component, where the depth component is represented by 8-bit grey scale images. Provided that the color and depth pair of the base view is encoded/decoded, a dependent view can be predicted from the base view via traditional translational block matching represented by the disparity vector. This process is often referred to as Disparity Compensated Prediction (DCP). As an alternative in this paper, the dependent view can also be predicted by warping the base view to its viewpoint pixel-by-pixel using the encoded/decoded base view texture and depth components as shown in Fig. 1 with dash lines. This process is only invoked between the base view and its dependent views within the same access unit (the same time instance). The technique is referred to as View Synthesis Prediction (VSP). With MVD as input, the decoder can render the intermediate views in a low-complexity fashion by selecting appropriate neighboring viewpoints and warping the selected view's texture and depth components to the target viewpoint.

III. View Synthesis Prediction

In this section, the basic concept of VSP is discussed. Generally speaking, for VSP, there are two types of image warping techniques, namely forward warping and backward warping, depending on the availability of the depth map of the current view. Forward warping generates the synthetic view when the depth map from the reference viewpoint is available. In a video coding scenario, that means the depth map from the reference viewpoint has to be encoded/decoded before encoding/decoding the texture component of the current view.

In particular, for each pixel S_r at a location X_r in the reference picture, the depth sample value d_r is known. Note that d_r has the following relationship with the actual distance value Z,

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

where Z_{near} and Z_{far} stand for the nearest and farthest depth of the current view.

Using the property of the triangular similarity, the disparity value D can be written as

$$D = f \cdot l / Z \quad (2)$$

where f is the camera focal length and l is the baseline distance. Therefore, the original point P in the 3D scene can be rendered at position X_c in the synthesized viewpoint with

$$X_c = X_r - D \quad (3)$$

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

$$S_c(X_c) = S_r(X_r) \quad (4)$$

Similar derivations can be obtained for backward warping using the depth of the view to be synthesized. Due to the texture first coding order constraint in HEVC-based 3D coding, the forward warping is used in our scheme. As a convention of forward warping, the warping process is conducted over all the pixel samples in the reference view. After all samples in the reference view are warped to the target view, there may be some vacant samples, which do not have corresponding warped values from the reference view, known as hole samples. Therefore, hole filling techniques are need to fill the vacant samples.

In contrast to the frame level VSP, to reduce the decoder computation complexity, a block level forward warping can be used by finding the possible reference regions. However, although the block level forward warping can reduce the decoder complexity in the sense that forward warping is invoked only for those blocks chosen as VSP, the encoder complexity is increased as the reference blocks may overlap each other, which leads to unnecessary increased complexity.

In this paper, consistent with our previous work [6], the forward warping is still used at the frame level. However, instead of simply adding the synthesized view in the reference list, a novel unified rate-distortion optimized VSP representation is proposed and discussed within the HEVC-based 3D coding framework.

IV. VSP using Skip and Merge Modes for HEVC-based 3D Coding

A. Merge Candidate List for Skip and Merge Modes

Similar with the HEVC 2D video coding, there are three types of modes for an inter slice, namely Skip mode, Merge mode and Inter mode in the HEVC-based 3D video coding. For each Coding Unit (CU), a Skip flag is coded to indicate whether the current CU uses Skip mode. If the current CU uses Skip mode, a number of parallel motion predictors $M = \{m_k | k = 0, 1, ..., 5\}$ are constructed as a merge candidate list for the Skip mode, which includes spatial neighboring motion predictors, temporal neighboring motion predictors and the inter-view motion predictor (only available in 3D video coding) [7][8] shown in Fig. 3. At the encoder, a merge index k is decided based on the rate-distortion cost

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

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 bits to code the merge index k. Note that there is neither motion vector difference nor residue transmitted for Skip mode.

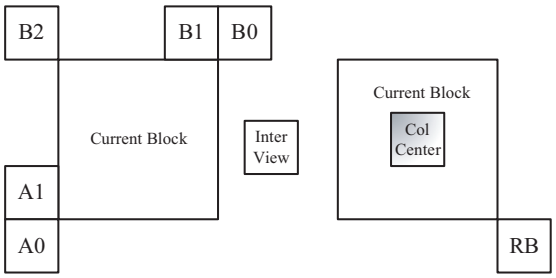

Fig. 3. Merge Candidate List Construction for HEVC-based 3D Coding

If the current CU is coded as Skip mode, the Prediction Unit (PU) partition within the CU is coded. Then for each PU, a merge flag is coded to indicate whether the current PU is Merge mode. If it is Merge mode, a merge candidate list for Merge mode is constructed similar as the merge list for Skip mode, where the merge index and the residue signals are coded and transmitted.

If the current PU is not coded as Merge mode, it is coded as a traditional Inter mode with prediction direction(bi- or uni- prediction), reference index, motion vector predictor index and residue signals coded.

B. VSP using Skip and Merge Candidate list

It can be found that the Skip and Merge modes were introduced according to the fact that motion can be directly derived from previously encoded information. Therefore, these modes, when signaled, could in effect represent the motion of a block without having to transmit other motion information required for Inter mode. Moreover, the motion predictor index coding can actually alleviate the increased overhead due to the inherent block based partition, when the neighboring block has the same motion as the current block. In practice, the Skip and Merge modes occupy a considerably large portion of modes in the bitstream, leading to an efficient block-based motion representation.

To efficiently represent the VSP mode, it is proposed to treat the VSP mode as a compensated prediction with a motion vector predictor included in the merge candidate list for Skip and Merge modes. Specifically, for VSP mode, the motion vector between the synthesized block and the current block is assumed to be (0,0) in both horizontal and vertical directions, since the synthesized block is a perfect match of the current block in theory by forward warping. Therefore, a motion vector predictor (0,0) referring to the synthesized frame is always included in the merge candidate list for Skip and Merge modes. That is, the merge candidate list is extended by adding (0,0) referring to the synthesized view from maximal 6 candidates to 7 candidates. Still, (5) is still used to evaluate the VSP mode against other compensated predictions to determine the best compensated prediction in terms of rate-distortion cost.

C. Candidate List Pruning

In the merge list, there is a considerably large probability that the neighboring blocks have the same motion information. If duplicate motion information exists in the candidate list, an unnecessary longer codeword (unary code for the merge list) is needed to represent the candidate, which deteriorates the coding performance. However, if comparisons are performed between every two candidates, it will result in an increased computation imposed on both the encoder and decoder. Therefore, in the proposed scheme, temporal and inter-view merge candidates are exempted from the pruning process, while VSP checking on the spatial neighbors is still performed.

In the current setting, the construction order within the merge list is Inter-view, A1, B1, B0, A0, VSP, B2, Temporal Col and Temporal Bottom-right. Note that, the actual generated list may vary according to the neighboring motion information. For VSP, comparisons are made against the spatial motion candidates. If a spatial candidate is VSP coded, VSP is not added additionally in the merge list. Therefore, the VSP index in the merge list can vary rather than a fixed position. In case of neighboring blocks A1 uses VSP and Inter-View is available, the codeword for VSP can be very short, more precisely, 2 bits.

V. Simulation Results

The proposed scheme is integrated into HTM3.1 [9] and the simulations are run under the common test condition defined in [10]. The performance is evaluated using excel embedded macro BDBR, where negative values stand for the bitrate saving against the anchor data. The coding structure is hierarchical B along the time axis, and IPP among base and dependent views, shown in Fig. 1. In the proposed scheme, prior to encoding/decoding each dependent view, a VSP generation process is invoked using the base view texture and depth to accomplish the forward warping and the synthesized view is regarded as an additional reference frame in the reference list. The proposed VSP scheme can be applied to both texture and depth.

The simulation results in Table I show that the proposed VSP scheme using Skip and Merge candidate list provides 3.7% bitrate reduction on average for the dependent view 1 and 1.9% bitrate reduction on average for the dependent view 2 with VSP applied to texture only. And 3.5% and 1.7% bitrate reduction is achieved with VSP applied to both texture and depth shown in Table II. It can be concluded that applying VSP would further improve the synthesized view quality.

Next, we study the VSP usage in the test sequences. Fig. 4(a) illustrates the VSP usage for the test sequence PoznanStreet of size 1920×1088 for the dependent view 1 at the anchor picture (base view intra coded). And Fig. 4(b) illustrates the VSP usage for dependent view 2. It can be observed that VSP is chosen in around 20%-30% areas within the frame. Also, from our experiments, we found that VSP is more frequently chosen in the anchor picture as no temporal references are available than in non-anchor pictures. The results suggest that VSP provides a good alternative predictor for anchor pictures in addition to the disparity compensation.

Another observation is that the VSP mode tends to be more frequently used in smooth regions rather than in the edge regions. The reason is that the synthetic view is generated using the depth information, and the depth-assisted warping will provide unreliable prediction for occluded areas, more precisely, the edge regions converged by foreground and background objects. Therefore, the predictor will probably have occlusion effects, resulting in a large residue.

VI. Conclusion

In this paper, an in-loop view synthesis framework is proposed, where the synthesized predictor is encoded as a special motion compensated predictor and the motion information is encoded as one of the motion predictors using skip/merge candidate list for HEVC-based 3D video coding. Apart from that, a merge list pruning process is used to reduce the code word length for VSP mode, which will further improve the coding performance of VSP mode. The proposed scheme is applicable to both texture coding and depth coding. The experimental results show that the proposed framework improved the coding performance up to 12.1% for dependent views. However, the decoder increased complexity is still an open issue to be addressed in order to strike a better balance between coding efficiency and complexity.

Acknowledgment

The authors would like to thank S. Shimizu from NTT for thoughtful discussion and cross verification of these results.

(a) (b)

Fig. 4. The usage of Prediction Unit (PU) chosen as VSP for HTM3.1 with VSP skip/merge added. 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 I

LUMA BD-RATE(%) OF THE PROPOSED VSP USING THE SKIP AND MERGE MODES FOR THE TEXTURE ONLY COMPARED WITH THE HTM3.1 ANCHOR.

Size	Sequence	video 0	video 1	video 2	video only	syn only	coded & syn	enc time	dec time
1024x768	Balloons	0.0	0.5	1.6	0.6	0.9	1.0	100.3%	157.3%
	Kendo	0.0	1.2	1.7	0.8	1.3	1.4	102.1%	157.9%
	Newspapercc	0.0	-0.1	0.0	0.1	-0.1	0.1	105.3%	185.6%
1920x1088	GhostTownFly	0.0	-7.2	-5.7	-1.4	-0.2	-0.3	104.3%	155.6%
	PoznanHall2	0.0	-2.7	1.2	0.0	0.8	0.8	108.2%	164.3%
	PoznanStreet	0.0	-5.4	-3.9	-1.3	-0.2	-0.3	107.4%	167.5%
	UndoDancer	0.0	-12.1	-7.9	-2.6	-1.3	-1.5	108.0%	164.0%
1024x768 Average		0.0	0.6	1.1	0.5	0.7	0.8	102.6%	166.4%
1920x1088 Average		0.0	-6.8	-4.1	-1.3	-0.2	-0.3	107.0%	162.8%
Average		0.0	-3.7	-1.9	-0.5	0.2	0.2	105.1%	164.3%

TABLE II

LUMA BD-RATE(%) OF THE PROPOSED VSP USING THE SKIP AND MERGE MODES FOR BOTH TEXTURE AND DEPTH COMPARED WITH THE HTM3.1 ANCHOR.

Size	Sequence	video 0	video 1	video 2	video only	syn only	coded & syn	enc time	dec time
1024x768	Balloons	0.0	0.7	1.7	0.7	0.8	1.0	111.9%	181.2%
	Kendo	0.0	1.3	1.8	0.9	1.0	1.2	111.2%	177.6%
	Newspapercc	0.0	0.0	-0.1	0.1	-0.1	0.1	119.8%	218.8%
1920x1088	GhostTownFly	0.0	-7.1	-5.6	-1.3	-1.5	-1.2	105.6%	182.5%
	PoznanHall2	0.0	-2.6	1.5	0.0	0.2	0.4	109.1%	193.6%
	PoznanStreet	0.0	-5.2	-3.6	-1.2	-0.4	-0.4	108.6%	208.8%
	UndoDancer	0.0	-11.8	-7.7	-2.5	-1.8	-1.8	109.7%	189.0%
1024x768 Average		0.0	0.7	1.1	0.5	0.6	0.8	112.0%	189.3%
1920x1088 Average		0.0	-6.7	-3.8	-1.3	-0.6	-0.6	109.1%	196.9%
Average		0.0	-3.5	-1.7	-0.5	-0.2	-0.1	110.8%	192.5%

REFERENCES

[1] 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, April 2011.

[2] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View Synthesis for Multiview Video Compression," in *Picture Coding Symposium (PCS)*, April 2006.

[3] S. Yea and A. Vetro, "Rd-optimized view synthesis prediction for multiview video coding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, 16 2007-oct. 19 2007, pp. I –209 –I –212.

[4] 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.

[5] 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.

[6] D. Tian, "CE1.a summary report: View synthesis and inter-view prediction," *Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCT2-A0011*, Stockholm, SE, July,2012.

[7] 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, July,2012.

[8] 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, JCT2-A0097*, Stockholm, SE, July,2012.

[9] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM 3.1.

[10] H. Schwarz and D. Rusanovskyy, "Common Test Conditions for 3DV Experimentation ," in *ISO/IEC JTC/SC29/WG11 MPEG, N12745,Geneva, Switzerland*, May 2012.