Simplified Depth-based Block Partitioning and Prediction Merging in 3D Video Coding

Fabian Jäger, Mathias Wien

Institut für Nachrichtentechnik, RWTH Aachen University
GERMANY

{jaeger,wien}@ient.rwth-aachen.de

***Abstract*—3D video is an emerging technology that bundles depth information with texture videos to allow for view synthesis applications at the receiver. Depth discontinuities define object boundaries in both, depth maps and the collocated texture video. Therefore, depth segmentation can be utilized for a fine-grained motion field partitioning of the corresponding texture component. In this paper, depth information is used to increase coding efficiency for texture videos by deriving an arbitrarily shaped partitioning. By applying motion compensation to each partition independently and eventually merging the two prediction signals, highly accurate prediction signals can be produced that reduce the remaining texture residual signal significantly. Simulation results show bitrate savings of up to 2.8% for the dependent texture views and up to about 1.0% with respect to the total bitrate.**

I. Introduction

3D video technology extends conventional stereo or multi view videos by adding corresponding depth maps. Emerging technologies, such as auto-stereoscopic and depth-adapting stereoscopic displays, require more than two views of the same scene to allow for their enhanced 3D capabilities. The Multi-View plus Depth (MVD) data format [1] bundles multiple texture videos with their corresponding depth maps and is designed for the aforementioned display technologies.

Consequently, combined coding of texture and depth video components becomes an important research topic with the goal to exploit inter-component dependencies for increasing overall coding performance. Using depth information to reduce texture bitrate seems more promising than the inverse, as the texture bitrate is typically significantly higher than the bitrate of the depth component.

Merkle et al. use reconstructed texture information to generate a segmentation mask, which is afterwards used for intra prediction of the collocated depth block [2]. For each of the two resulting segments a DC prediction value is coded. Reusing already coded motion information of the texture view to reduce the required bitrate of the same view's depth component is proposed in [3]. In the approach of Winken et al. motion vector information and also partitioning of the prediction units is inherited from the collocated texture block when coding the depth block. Jung and Mora propose to limit the depth of the coding quad-tree of the depth map to the corresponding texture quad-tree [4]. This limitation allows to save some bitrate for splitting flags in the depth component, but at the same time it introduces a parsing dependency between the two components.

All three approaches utilize texture information to improve coding efficiency of the depth map. The resulting bitrate reduction with respect to the overall bitrate of the 3D video bitstream is relatively low. This is due to the fact that the depth bitrate only accounts for approx. 9-10% of the overall bitrate. Consequently, exploiting depth information to code the texture component more efficiently seems the more promising approach. Synthesizing a prediction signal for the dependent texture views based on coded depth information is proposed by Lee et al. in [5] and by Jger et al. in [6]. View synthesis prediction (VSP) is an efficient way for reducing the required bitrate of dependent texture views, but it introduces additional complexity to the decoder due to its fine-grained disparity compensation, including irregular memory access to the decoded picture buffer.

In [7], it was further proposed to utilize reconstructed depth information to derive a binary segmentation mask, which allows to independently predict foreground and background, as depicted in Figure II. Thus, the texture block's motion field can be approximated at pixel precision without splitting it into many small sub-blocks as with VSP. In that publication it was assumed that a texture block's collocated depth map block is already decoded and reconstructed, which is a restriction to the bitstream as it requires depth maps to be coded before the collocated texture view.

The coding tool proposed in this paper extends the concept of [7] by allowing a texture-first coding configuration and by significantly reducing the complexity for deriving the texture block's motion partitioning. It was recently proposed to JCT-3V and adopted to the latest working draft for the standardization of a 3D extension to HEVC [8].

The remainder of this paper is structured as follows: Section II describes the overall concept of Depth-based Block Partitioning (DBBP) whereas Section III explains the bitstream signaling of the resulting data. Experimental results based on the proposed algorithm are presented in Section IV. Finally, Section V summarizes the results of the proposed coding

978-1-4799-6139-9/14/$31.00

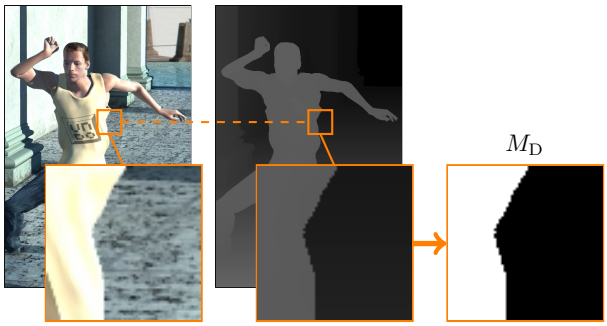

Fig. 1. Cropped frame from *Undo_Dancer* with magnified component blocks of a coding unit. The binary segmentation mask M_{D} is derived from the collocated depth block.

method and gives an outlook on potential further research activities.

II. Block Partitioning Algorithm

The proposed algorithm for using reconstructed depth information to predict the partitioning of texture coding units consists of four sequential steps, which are shortly summarized in the following section.

A. Virtual Depth Derivation

In a 3D video coding bitstream, texture information is typically coded before the corresponding depth component. In such a configuration, the collocated depth block is not yet available when decoding the associated texture coding unit. To overcome this limitation, Zhang et al. proposed to use a neighboring block's coded disparity vector (NBDV) to derive the location of the corresponding depth block in the base view's depth map [9], [10]. By this approach, a virtual depth block for the dependent views' texture component can be fetched from the completely reconstructed base view and used for further processing. The proposed coding tool makes use of this virtual depth map in configurations where texture is coded before depth. In the remainder of this paper such a texture-first coding order is assumed, as it is also defined by the JCT-3V common test conditions [11].

B. Depth Segmentation

In a second step, the virtual depth block of the current texture component is segmented into two arbitrarily shaped segments. The segmentation is performed based on a simple thresholding mechanism. The threshold $\bar{d}$ is computed as the mean of the four corner pixels of the virtual depth map.

$$\bar{d} = \frac{1}{4}\Big[d(0,0) + d(0,2N-1) + d(2N-1,2N-1) + d(2N-1,0)\Big] \quad (1)$$

Here, $2N$ defines the edge length of the current square texture block and $d(x,y)$ resembles a sample at position x,y of the virtual depth map D, fetched from the base view's reconstructed depth map. In comparison to the approach in [7] this already reduces the computational complexity from $(2N)^2$ to 3 additions per coding unit.

Afterwards, a binary segmentation mask M_{D} of size $2N\times 2N$ is derived based on $\bar{d}$ as follows.

$$m_{\mathrm{D}}(x,y) = \begin{cases} 1, & \text{if } d(x,y) \geq \bar{d}, \\ 0, & \text{otherwise.} \end{cases}, x,y \in [0, 2N-1] \quad (2)$$

The resulting mask describes the location of foreground and background objects within the current block. While motion or disparity compensation in a modern video codec (e.g. in HEVC) is performed on rectangular (sub-)blocks, an arbitrarily shaped compensation typically requires pixel-wise processing. Such a concept was originally applied in view-synthesis prediction (VSP), which warps individual pixels or very small pixel groups based on the corresponding depth value to the position in the particular reference view. Higher order deformations can be approximated by this fine-grained compensation. At the same time it introduces relatively high computational complexity compared to conventional block-based motion/disparity compensation. This is mainly due to irregular memory accesses to a reference buffer and pixel-wise conversion from depth to disparity.
The proposed simplified depth-based block partitioning (DBBP) scheme solves this trade-off problem. It still uses block-based compensation (at full block size) in the prediction stage, but allows pixel-precise approximation of motion/disparity discontinuities due to its segmentation mask.

C. Segment-wise Compensation

In the proposed DBBP scheme, the actual motion or disparity compensation is performed on a $2N\times 2N$ partitioning. This full-size motion/disparity compensation is performed twice, once for each segment, and results in two prediction signals $P_{\mathrm{T}0}$ and $P_{\mathrm{T}1}$. This process can be interpreted as a weighted prediction with two prediction signals (bi-prediction) with a binary mask for the weighting.
Consequently, two sets of motion/disparity information need to be coded for a DBBP block. The assumption behind this approach is that a texture block can typically be segmented into foreground and background based on the collocated depth block. These two depth layers can be best compensated independently by their own sets of motion or disparity vectors. A conventional video codec would need to approximate the same motion discontinuity by splitting the region into very small blocks that need to be compensated individually. The proposed method allows to retain bigger block sizes and derive the motion discontinuity from the virtual depth information.

D. Merging of Prediction Signals

After having generated two full-size prediction signals $P_{\mathrm{T}0}$ and $P_{\mathrm{T}1}$ for a DBBP block, the segmentation mask M_{D} is used to merge these into the final prediction signal P_{T} for the current texture block, as depicted in Figure 3 and defined in

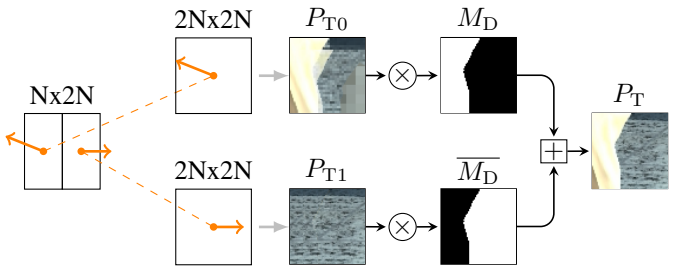

Fig. 2. Merging process: For each of the two partitions a $2N\times2N$ motion compensation is performed. The prediction signals P_{T0} and P_{T1} are combined using the mask M_D.

the following equation.

$$p_T(x,y) = \begin{cases} p_{T0}(x,y), & \text{if } m_D(x,y) = 0, \\ p_{T1}(x,y), & \text{otherwise.} \end{cases}, x,y \in [0, 2N-1] \quad (3)$$

By merging the two prediction signals, shape information from the depth map allows to independently compensate foreground and background objects within the same texture block. At the same time, DBBP does not require pixel-wise motion/disparity compensation. Thus, memory accesses to the reference buffers are always regular (block-based) in contrast to approaches like VSP. This is preferable with respect to computational complexity, because of a higher probability for finding the required reference pixels in the memory cache.

III. Signaling of Motion Data

As described in the previous section, DBBP requires coding of two sets of motion information, one for each segment. A modern video coder, such as HEVC, allows to use rectangular, non-square partitioning modes within a coding unit (CU) for fine-grained motion compensation. For each of these partitions in a CU a separate set of motion information is coded. This coding scheme is reused for depth-based block partitioning.

After the encoder has derived the optimal motion/disparity information for each DBBP segment, this information is mapped into one of the available rectangular, non-square partitioning modes of HEVC, including asymmetric motion partitioning modes [12]. The mapping of the binary segmentation mask to one of the six available two-segment partitioning modes is performed by a low complex sample-based scheme, which requires only up to three comparisons instead of a full block correlation analysis, as it was proposed in [7]. To find the best matching partitioning mode k_{opt} for a given M_D, the algorithm illustrated in Figure 3 is performed. The complexity reduction compared to [7] is illustrated in Table I. While the number of operations in the original partitioning derivation scheme depends on the size of the processed coding unit, the proposed method requires a constant number of operations to achieve approximately the same results, as can be seen in Section IV.

After having found the best matching conventional partitioning mode, motion information is stored according to this optimal mode k_{opt}. Succeeding coding units (CUs) can access

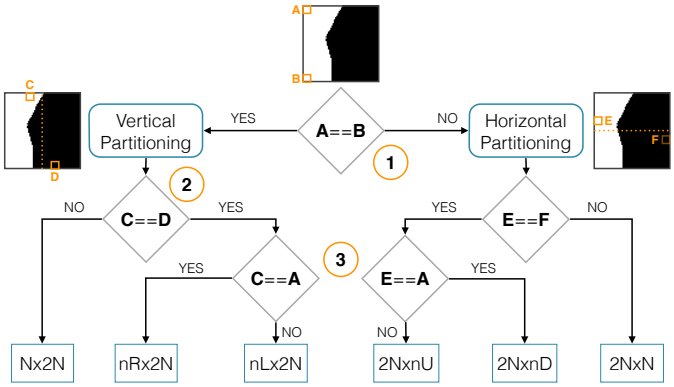

Fig. 3. Partitioning derivation process based on segmentation mask M_D, which only requires up to three comparisons.

the already coded motion information conventionally when deriving motion vector candidates for advanced motion vector prediction or motion vector merging.

A single flag is required for signaling the usage of DBBP for prediction. To avoid sending this flag for each partitioning mode, the partitioning mode for DBBP-coded blocks is temporarily set to $2N\times N$ before encoding. Consequently, the DBBP flag needs to be coded only for this partitioning mode. The decision to select $2N\times N$ partitioning for hiding the DBBP flag is based on an analysis of the usage of all of the existing partitioning modes. As $2N\times N$ is used least, it is an obvious choice to hide an additional flag in that particular mode.

At the decoder, the true partitioning is derived for DBBP blocks based on the algorithm described in Figure 3. All further parsing steps remain the same as in the HEVC base specification. It is to be noted that this derivation process does not introduce a parsing dependency on another slice. As the derivation process always results in a partitioning with two prediction units, the rest of the syntax parsing does not depend on the result of the derivation.

IV. Experimental Results

The proposed depth-based block partitioning algorithm was implemented into the JCT-3V test model (HTM 9.0r1) [13]. The presented simulation results are performed according to the JCT-3V common test conditions [11].

The simulation results in Table II clearly show the benefits of depth-based block partitioning with respect to coding efficiency. While the bitrate reduction for the dependent views gets up to 2.8%, the resulting bitrate reduction w.r.t. the total bitrate finds its maximum at about 1.0%. This effect can be explained by the fact that the bitrate of the base view remains unchanged when using DBBP as it is only applied to dependent views to retain an HEVC-compatible base view. Moreover, it can be seen that the bitrate reduction varies between different test sequences, which is mainly due to their varying quality in reconstructed depth maps. Whenever the segmentation mask from the reconstructed depth map is aligned with object boundaries in the texture component,

TABLE I
COMPLEXITY COMPARISON BETWEEN THE PARTITIONING DERIVATION METHOD IN [7] AND THE PROPOSED SCHEME.

Block Size	Original Method [7]				Proposed Method		
	Thresholding		Partitioning Derivation		Thresholding		Partitioning Derivation
8×8	3 ADDs	1 SHIFT	24+18 COMPs	24 ADDs	3 ADDs	1 SHIFT	< 3 COMPS
16×16	15 ADDs	1 SHIFT	96+18 COMPs	96 ADDs	3 ADDs	1 SHIFT	< 3 COMPS
32×32	63 ADDs	1 SHIFT	384+18 COMPs	384 ADDs	3 ADDs	1 SHIFT	< 3 COMPS
64×64	255 ADDs	1 SHIFT	1536+18 COMPs	1536 ADDs	3 ADDs	1 SHIFT	< 3 COMPS

DBBP yields a fine-grained motion separation between foreground and background without further splitting of the coding unit. The complexity reduction proposed in this paper does not have an impact on the coding performance of DBBP while it significantly reduces the required number of operations.

V. SUMMARY

This paper proposes a significant complexity reduction of the depth-based block partitioning (DBBP) coding scheme while enabling DBBP for texture-first coding configurations by using a virtual depth map. With depth-based block partitioning (DBBP) reconstructed depth information is used to derive a segmentation mask, which thereafter allows for an independent, fine-grained motion/disparity compensation of foreground and background. The derived segmentation mask is eventually utilized to merge the two compensated blocks. DBBP is not relying on the correctness of actual depth values, as long as segment information in texture and depth are aligned.

The proposed algorithm can reduce the bitrate of the two dependent views by approximately 1.3% and 0.9%, respectively. For some sequences the bitrate reduction for the dependent views is even at up to 2.8% and 2.5%. The proposed significant complexity reductions do not have an impact on this coding efficiency.

Further research is needed to investigate how joint encoder optimization of texture and depth may help the proposed scheme. As the depth component is currently optimized independently, some information about the location of depth discontinuities might get lost during optimization. Moreover, it needs to be investigated whether the compensation aspect of DBBP can replace the sub-block prediction process of VSP while retaining the same prediction quality.

TABLE II
BD-RATE SAVINGS WHEN EXTENDING HTM 9.0R1 WITH THE PROPOSED DEPTH-BASED BLOCK PARTITIONING SCHEME.

Sequence	Texture View 1	Texture View 2	Total Texture
Balloons	-1.0 %	-0.1 %	-0.4 %
Kendo	-0.7 %	-0.4 %	-0.3 %
Newspaper_CC	-0.8 %	-0.7 %	-0.3 %
GT_Fly	-1.3 %	-0.7 %	-0.4 %
Poznan_Hall2	-0.3 %	-0.4 %	-0.3 %
Poznan_Street	-1.1 %	-1.5 %	-0.6 %
Undo_Dancer	-2.8 %	-2.5 %	-1.0 %
Shark	-2.1 %	-1.2 %	-0.5 %
1024x768	-0.8 %	-0.4 %	-0.3 %
1920x1088	-1.5 %	-1.2 %	-0.6 %
AVERAGE	-1.3 %	-0.9 %	-0.5 %

REFERENCES

[1] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, 2007, pp. 201–204.

[2] P. Merkle, C. Bartnik, K. Müller, D. Marpe, and T. Wiegand, "3D video: Depth coding based on inter-component prediction of block partitions," in *Proceedings of IEEE Picture Coding Symposium*, Kraków, Poland, May 2012, pp. 149–152.

[3] M. Winken, H. Schwarz, and T. Wiegand, "Motion vector inheritance for high efficiency 3D video plus depth coding," in *Proceedings of IEEE Picture Coding Symposium*, Kraków, Poland, 2012, pp. 53–56.

[4] Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, "3D-CE3.h: Depth quadtree prediction for 3DHTM 4.1," JCT3V-B0068, Tech. Rep., October 2012.

[5] C. Lee and Y.-S. Ho, "A framework of 3D video coding using view synthesis prediction," in *Proceedings of IEEE Picture Coding Symposium*, Kraków, Poland, 2012, pp. 9–12.

[6] F. Jäger and C. Feldmann, "Warped-skip mode for 3D video coding," in *Proceedings of IEEE Picture Coding Symposium*, Kraków, Poland, 2012, pp. 145–148.

[7] F. Jäger, "Depth-based block partitioning for 3D video coding," in *Proc. of International Picture Coding Symposium PCS '13*. San Jose, USA: IEEE, Piscataway, Dec. 2013.

[8] F. Jäger, J. Konieczny, and G. Cordara, "CE3: Results on depth-based block partitioning (DBBP)," Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, 7th Meeting, San Jose, USA, Doc. JCT3V-G0106, Jan. 2014.

[9] L. Zhang, Y. Chen, and M. Karczewicz, "Disparity vector based advanced inter-view prediction in 3D-HEVC," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, 2013, pp. 1632–1635.

[10] L. Zhang, J. Kang, X. Zhao, Y. Chen, and R. Joshi, "Neighboring block based disparity vector derivation for 3D-AVC," in *Visual Communications and Image Processing (VCIP), 2013*, 2013, pp. 1–6.

[11] Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, "Common test conditions of 3DV core experiments," JCT3V-G1100, San Jose, USA, Tech. Rep., January 2014.

[12] I.-K. Kim, S. Lee, M.-S. Cheon, T. Lee, and J. Park, "Coding efficiency improvement of HEVC using asymmetric motion partitioning," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2012 IEEE International Symposium on*, Seoul, June 2012, pp. 1–4.

[13] Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T VCEG and ISO/IEC MPEG, "Test model 7 of 3D-HEVC and MV-HEVC," Doc. JCT3V-G1005, San Jose, USA, Tech. Rep., January 2014.