Segment-wise Prediction in 3D Video Coding

Fabian Jäger
Institut für Nachrichtentechnik
RWTH Aachen University, GERMANY

***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 segmentation of the corresponding texture component.**

In this paper, depth information is used to increase coding efficiency for texture videos by deriving an arbitrarily shaped segmentation mask. By applying independent motion compensation to each of these segments and subsequently merging the two prediction signals, a highly accurate prediction signal can be generated that reduces the energy in the remaining texture residual signal. 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 video by the addition of depth information. Emerging display technologies, such as auto-stereoscopic and depth-adapting stereoscopic displays, require more than two views of the same scene to enable 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, joint coding of texture and depth information becomes an important research topic with the goal to exploit inter-component dependencies for increasing overall coding performance. Utilizing auxiliary depth information to reduce texture bitrate seems more promising than the inverse approach, as the texture bitrate is typically significantly higher than the bitrate of the depth component, as depicted in Figure 1. The importance of the development of an efficient coding scheme for 3D video input data (including texture and depth videos) is further reflected in recent standardization activities in the scope of JCT-3V. This group currently works on a 3D extension to the latest video coding standard, HEVC.

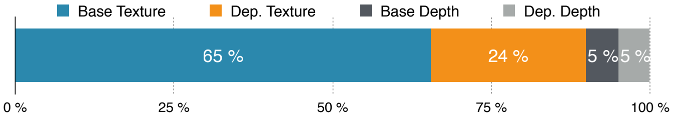

Fig. 1: Average bitrate distribution of the sequences coded according to the JCT-3V common test conditions [2].

In this scope, Merkle et al. propose to use reconstructed texture information to derive a segmentation mask, which is afterwards used for more precise intra prediction of the collocated depth block [3]. For each of the two resulting segments of the depth map a DC offset value is coded to further improve the prediction signal. Reusing already coded motion information of the texture view to reduce the required bitrate of the same view's depth component is further proposed in [4]. In this approach of Winken et al., motion vector information and also partitioning into prediction units is inherited from the collocated texture block when coding a depth block. Jung and Mora propose to limit the depth of the coding quad-tree of the depth map to not be deeper than the quad-tree of the corresponding texture view [5]. This limitation allows to save some bitrate for otherwise required splitting flags in the depth component, but at the same time it introduces an undesired parsing dependency between the two components.

All mentioned approaches utilize texture information to improve coding efficiency of the collocated 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, as shown before. Consequently, exploiting depth information to code the texture component more efficiently seems more promising. Synthesizing a prediction signal for the dependent texture views based on coded depth information is proposed by Lee et al. in [6] and also by us in [7]. So-called View synthesis prediction (VSP) is an efficient way of reducing the required bitrate of dependent texture views, but it introduces additional computational complexity to the decoder due to its fine-grained disparity compensation, including irregular memory access to the decoded picture buffer.

In [8], we proposed to utilize already decoded depth information to derive the partitioning of the collocated texture component into prediction units. This approach allows to save some signaling overhead for the partitioning mode. We further proposed a method to better approximate the motion or disparity field discontinuities of the texture component by incorporating segmentation information from the corresponding depth map.

In this paper, the concept of [8] is further generalized by extending the way of signaling the usage of segment-wise prediction (SP) and by allowing SP for all possible coding configurations. It was recently proposed to JCT-3V and is currently further investigated in the scope of this standardization activity.

The remainder of this paper is structured as follows: Section II describes the overall concept of segment-wise prediction whereas Section III explains different ways of signaling the usage of SP to the decoder. Experimental results based on the proposed algorithm are presented in Section IV. Finally, Section V summarizes the results of the proposed coding method and gives an outlook on potential further research activities.

II. Segment-wise Prediction Concept

The whole algorithm of segment-wise prediction consists of four major steps, which are the fetching of the depth block (1), depth segmentation (2), segment-wise compensation (3) and the final prediction merging (4). The following sections describe each step individually.

A. Fetching of a Collocated Depth Block

The proposed segment-wise prediction mechanism requires to derive the segmentation mask from the collocated depth map. Thus, this depth block needs to be fetched in a very first step. As 3D video coding does not impose any restrictions on the coding order of texture and depth components, a texture view's corresponding depth component is not necessarily available when decoding the texture. To still be able to employ SP, it is 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]. This depth block can be used for the current position in the dependent view as it describes exactly the same objects, shifted by a disparity vector relative to the base view's camera position. In case of a depth-first coding order for the dependent views, the depth block fetching is simplified as the collocated depth information is already decoded and reconstructed and can be used directly.

B. Depth Segmentation

In the second step of segment-wise prediction, the previously fetched depth block is segmented into two arbitrarily shaped segments. The segmentation is performed based on a simple thresholding mechanism to keep the required computational complexity low. The threshold $\bar{d}$ is computed as the mean of the four corner pixels of the depth map block.

$$\begin{aligned}\bar{d} = \frac{1}{4}\Big[&d(0,0) + d(0,2N-1)\\ &+ d(2N-1,2N-1) + d(2N-1,0)\Big]\end{aligned} \tag{1}$$

Here, $2N$ denotes the edge length of the current square texture block (and the fetched depth block of the same size) and $d(x,y)$ resembles a sample at position x, y within the depth map block D. Afterwards, a binary segmentation mask M_D of size $2N{\times}2N$ is derived based on $\bar{d}$ as follows.

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

The resulting mask M_D defines the location of foreground and background objects within the current texture/depth block. Motion or disparity compensation in a modern video codec (e.g. in HEVC) is performed on rectangular partitions to avoid arbitrarily shaped compensation, which typically requires pixel-wise processing. Such a pixel-wise prediction mechanism was first employed with view-synthesis prediction (VSP), which warps individual pixel positions or very small regions based on the corresponding depth value to the position in the particular reference view. Higher order deformations can be better approximated by this very 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 segment-wise prediction (SP) scheme solves this trade-off problem. It still uses block-based compensation (at full block size) with regular memory access 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 SP scheme, the actual motion or disparity compensation is performed on the full block without any sub-partitioning. This full-size motion/disparity compensation is performed twice, once for each segment, and results in two prediction signals P_T0 and P_T1. This process can be interpreted as a weighted bi-prediction with two prediction signals (as already employed in HEVC), but with a binary mask for the weighting. Therefore, it does not increase worst-case complexity compared to bi-prediction, which only needs to merge two individual prediction signals.

Consequently, two sets of motion/disparity information need to be coded for an SP block. The assumption behind this approach is that conventional block-based motion compensation fails along motion discontinuities, which themselves occur along object boundaries. As the already coded depth information holds information about object boundaries, segment-wise prediction utilizes this knowledge to improve the prediction quality for the collocated texture component. The two resulting segments can be better 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 sub-blocks that need to be compensated individually. The proposed method allows to retain bigger block sizes and derive the location of motion discontinuities from the collocated depth block.

D. Merging of Prediction Signals

After having generated two full-size prediction signals P_T0 and P_T1 for an SP block, the previously derived segmentation mask M_D is used to merge these into the final prediction signal P_T for the current texture block. This process is depicted in Figure 3 and defined in the following equation.

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

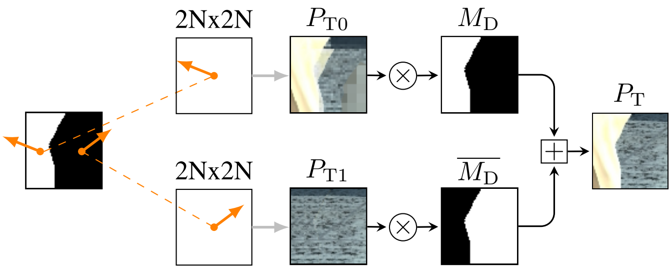

Fig. 2: Merging process: For each of the two segments, a $2N{\times}2N$ motion compensation is performed. The resulting prediction signals P_{T0} and P_{T1} are combined using the mask M_{D}, which was derived from the collocated depth block.

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. Nevertheless, SP 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 of finding the required reference sample values in the memory cache.

III. Signaling of Segment-wise Prediction Mode

As described in the previous section, SP 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 an approximation of fine-grained motion compensation. For each of these two partitions in a CU, a separate set of motion information is coded. This coding scheme is reused for the proposed segment-wise prediction mechanism.

After the encoder has derived the optimal motion/disparity information for each SP segment, the resulting two sets of motion information are signaled the same way as a $2N \times N$ partitioning in HEVC would signal the two individual motion vectors. In addition to the actual motion information, the decoder further needs to know that the described SP prediction mechanism is to be applied instead of the conventional PU-based method. In the proposed scheme, two alternative ways are defined: As the first option, a single flag is transmitted for each coding unit explicitly telling the decoder to apply SP when performing the prediction for the current CU. This approach introduces additional overhead information to be coded for each CU, also for coding units that are not using the proposed prediction scheme. Alternatively, it is proposed to add a special *SP merge candidate* to the list of merge candidates, which is constructed for each prediction unit. If the signaled merge index selects the *SP merge candidate* to be used for the current PU, the aforementioned process is performed at the decoder. This new merge candidate is only available for square $2N \times 2N$ prediction units to not overlap with the explicit flag, which is signaled at the CU level (in case of non-square CU partitioning) and which explicitly sends two sets of motion information. In case of the newly added merge candidate, the two sets of motion information need to be derived according to the following derivation rule. Only if both segments can be assigned a set of motion information, SP is available as a merge candidate.

- Check conventional neighboring merge candidates (A_i, B_i and C_i) for availability (in fixed order) and assign their motion vector to either segment 0 or segment 1, according to the closest segmentation mask corner value.
- If at least one set of motion information for both segments is found, add segment prediction (SP) merge candidate to the merge candidate list.
- If signaled merge index selects SP merge candidate, apply SP prediction mechanism to PU, as described in Section II.

The first item of this derivation process of the two sets of motion information is further depicted in Figure 3. The primary advantage of the signaling of SP by a special merge candidate is that only the corresponding merge index needs to be signaled in the bitstream to enable segment-wise prediction for a $2N \times 2N$ prediction unit. The two sets of motion information are implicitly derived at the decoder from neighboring motion.

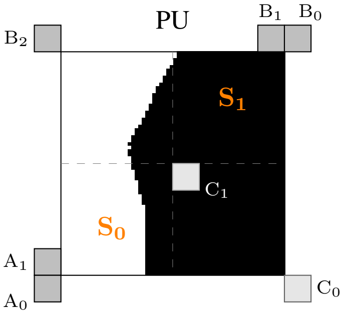

Fig. 3: For the SP merge candidate, the conventional merge candidate positions of a $2N \times 2N$ prediction unit (PU) are checked and if available, their motion vectors are assigned to one of the two segments, based on the depth-derived segmentation mask.

IV. Experimental Results

The proposed segment-wise prediction algorithm was implemented into the JCT-3V test model (HTM 11.0) [11]. The presented simulation results in Table I are performed according to the JCT-3V common test conditions [2] while Table II show results when using a depth-first coding configuration for the dependent views (for the reference and for the modified HTM).

The simulation results in Table I clearly show the benefits of the proposed segment-wise prediction algorithm with respect to coding efficiency improvements. While the bitrate reduction for the dependent views gets up to 2.8%, the resulting bitrate reduction relative to the total bitrate finds its maximum at about 1.0% (*Undo_Dancer*). This deviation can be explained by the fact that the bitrate of the base view remains unchanged because SP is only applied to the dependent views to retain

TABLE I: BD-Rate savings when adding segment-wise prediction to HTM 11.0 with a texture-first coding configuration.

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 %**

TABLE II: BD-Rate savings when adding segment-wise prediction to HTM 11.0 with a depth-first coding configuration.

Sequence	**Texture View 1**	**Texture View 2**	**Total Texture**
Balloons	-1.2 %	-0.8 %	-0.6 %
Kendo	-0.8 %	-0.6 %	-0.4 %
Newspaper_CC	-0.9 %	-0.8 %	-0.4 %
GT_Fly	-1.6 %	-0.9 %	-0.5 %
Poznan_Hall2	-0.8 %	-0.7 %	-0.4 %
Poznan_Street	-1.8 %	-1.9 %	-0.8 %
Undo_Dancer	-3.6 %	-3.1 %	-1.4 %
Shark	-2.9 %	-2.2 %	-0.9 %
1024x768	**-0.9 %**	**-0.7 %**	**-0.4 %**
1920x1088	**-2.1 %**	**-1.8 %**	**-0.8 %**
AVERAGE	**-1.7 %**	**-1.4 %**	**-0.7 %**

an HEVC-compatible base layer.

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, SP yields an accurate motion separation between foreground and background without the necessity of further splitting of the coding unit.

When employing a depth-first coding configuration, SP does not need to fetch blocks from the base view's depth map for the segmentation mask derivation. Rather, it can use the more accurate depth information from the same view as the current texture CU. This slight change impacts the resulting compression efficiency, as can be seen in Table II. Especially for sequences with more accurate depth maps, the coding gain is further increased.

V. Conclusion

This paper proposes to utilize reconstructed depth information to derive a segmentation mask, which thereafter allows for an independent, fine-grained motion/disparity compensation of foreground and background objects within a coding unit. SP is not relying on the magnitude or precision of actual depth values (as VSP does), 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%. This bitrate reduction can be further increased if using a depth-first coding configuration.

Further research is needed to investigate how joint encoder optimization of texture and depth may help the improve the proposed prediction method. 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 SP can replace the sub-block prediction process of VSP while retaining the same or similar prediction quality.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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, "Test model 7 of 3D-HEVC and MV-HEVC," Doc. JCT3V-G1005, San Jose, USA, Tech. Rep., January 2014.