3D Video Coding with Depth Modeling Modes and View Synthesis Optimization

Karsten Müller*, Philipp Merkle*, Gerhard Tech* and Thomas Wiegand*,†
*Image Processing Department, Fraunhofer HHI, Berlin, Germany
†School of EE & CS, Berlin Institute of Technology, Berlin, Germany

Abstract— **This paper presents efficient coding tools for depth data in depth-enhanced video formats. The method is based on the high-efficiency video codec (HEVC). The developed tools include new depth modeling modes (DMMs), in particular using non-rectangular wedgelet and contour block partitions. As the depth data is used for synthesis of new video views, a specific 3D video encoder optimization is used. This view synthesis optimization (VSO) considers the exact local distortion in a synthesized intermediate video portion or image block for the depth map coding. In a fully optimized 3D-HEVC coder, VSO achieves average bit rate savings of 17%, while DMMs gain 6%in BD rate, even though the depth rate only contributes 10% to the overall MVD bit rate.**

I. INTRODUCTION

3D video formats have been extended from stereoscopic video to multi-view video plus depth (MVD) in order to support different stereoscopic as well as multi-view displays [1][8] with one generic format. In particular, MVD formats use only 2 or 3 original camera views associated with per pixel depth or disparity information. At the receiver side additional required views for the 3D display are generated by means of depth-image-based rendering (DIBR). For such depth-enhanced video formats, efficient 3D video coding solutions are currently developed, e.g. in the Joint Collaborative Team on 3D Video Coding Extension Development, formed by ISO-MPEG and ITU-VCEG.

The state-of-the-art standard for multi-view and 3D video compression is the MVC extension of H.264/MPEG-4 AVC [6], which has also been adopted by Blu-ray. Originally, MVC was developed as a coding solution for multi-view video data by introducing inter-view prediction [3][11] to AVC simulcast in order to exploit the similarities between input views obtained from slightly different camera positions. For MVD coding, a simple extension is to apply two MVC codecs: One for all video and one for all depth data. MVC however was optimized for video data, and the characteristic of depth signals is different. Usually, depth maps contain larger areas of nearly constant or only slowly varying values within objects, as well as sharp edges at foreground/background object boundaries.

Therefore, improved coding approaches are currently investigated, that are based on high-efficiency video coding (HEVC) for 2D video data [18]. Similar to the MVC extension of H.264/AVC, inter-view prediction was introduced to HEVC [17] in order to exploit similarities between different views. An extension to inter-view prediction is view synthesis prediction, where depth is used to warp an image block into other views for better matching [21]. In MVD, also inter-component dependencies between video and depth data of the same view exist, e.g. similar motion vectors. Accordingly, specific depth coding tools have been developed, which inherit such data from the video components [14][20]. To account for the different characteristics of depth signals, specific depth coding methods are required. Therefore, we developed tools for edge preservation in depth maps, based on wedgelet and contour modeling. They provide non-rectangular block partitioning in areas with important depth edges. Wedgelet and contour partitioning can be derived within the depth data, as well as inherited from the video data by inter-component prediction.

Wedgelet- and platelet-based coding of depth images was shown by Morvan et al. [12], where depth blocks were modeled by piecewise linear functions without using any residual data. This method was further tested for MVD sequences by Merkle at al. [10], showing that it requires a 25% higher bit rate compared to H.264/AVC intra-only coding at the same objective quality in terms of PSNR. However, better visual quality was shown, when intermediate views where rendered with platelet-based depth map coding at the same bit rate. Wedgelets were also used by Ferreira et al. [4] for geometric partition-based motion vector prediction in video coding to gain higher coding efficiency. Finally, the encoding process for depth maps needs to be adapted, since depth maps are used for synthesizing intermediate video views and thus not directly viewed. For this, methods for estimating the synthesized view distortion depending on distorted depth data have been presented and used in encoding by Kim [7] and Oh [15].

In this paper, we present the developed depth coding for MVD data, which uses wedgelet- and contour-based depth modeling modes together with view synthesis optimization, as follows: In Section II, the depth coding methods are presented, including modeling modes with four different options of depth edge position signaling are explained. Then, the view synthesis optimization for depth encoding with its specific distortion measure calculation is introduced. Coding results for our approach are shown in Section III, while conclusions are drawn in Section IV.

II. DEPTH CODING METHODS

For depth map coding, the special characteristic and purpose of this information is considered with unstructured constant or slowly changing areas for scene objects and abrupt value changes at object boundaries between foreground and background areas. Experiments with state-of-the-art compression technology have shown that such depth maps can be compressed very efficiently. In addition, sub-sampling to a lower resolution prior to encoding and decoder-side up-sampling similar to chrominance sub-sampling has also been studied with good results [16]. Since the purpose of depth maps is to provide scaled disparity information for texture data for view synthesis, coding methods have to be adapted accordingly. Especially the sharp depth edges between foreground and background areas should be preserved during coding. A smoothing of such edges, as done by classical block-based coding methods, may lead to visible artifacts in intermediate views [13]. Furthermore, depth coding has to be optimized with respect to the quality of synthesized views, as the quality of the reconstructed depth data is irrelevant. For this, the rate-distortion optimization is adapted for depth data at the encoder. As depth data is used for view synthesis of the associated texture information, the distortion measure for depth encoding is obtained by comparing uncoded synthesized and reconstructed synthesized views.

A. Partition-based Depth Coding

For a better preservation of edge information in depth maps, we developed wedgelet and contour-based modeling modes. During encoding, each depth block is analyzed for significant edges. If such an edge is present, a block is subdivided into two non-rectangular partitions $\mathbf{P}_1$ and $\mathbf{P}_2$ as shown in Fig. 1.

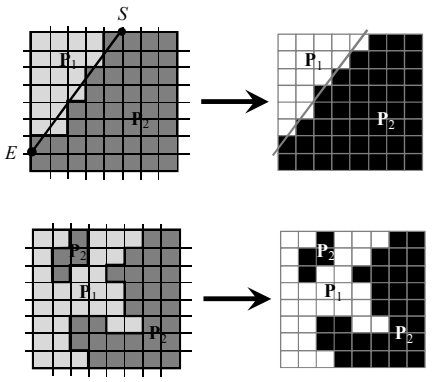

Fig. 1 Wedgelet partition (top) and contour partition (bottom) of a depth block: Original sample assignment to partitions P_1 and P_2 (left) and partition pattern (right).

The partitions can be separated by a straight line as an approximation of a rather regular depth edge in this block (see Fig. 1 top). Both partitions are then represented by a constant value. In addition to these values, the position of the separation line or contour is encoded in different ways:

1) An explicit signaling is carried out, using a look-up table. This table contains all possible separation lines within a block in terms of position and orientation, and provides an index for them.
2) A separation line can also be derived from neighboring blocks, e.g. if an already coded neighboring block contains significant edge information which end at the common block boundary. Then, a continuation of this edge into the current block can be assumed. Accordingly, one or both end points of the separation line (S and E in Fig. 1, top left) can be derived from the already coded upper and left neighboring block, and thus don't need to be signaled.
3) The position of a separation line can be derived from the corresponding texture block.
4) If a depth block contains a more complex separation between both partitions, as shown in Fig. 1 bottom, its contour can also be derived from the corresponding texture block.

In the depth encoding process, either one of the described depth modeling modes with signaling of separation information and partition values, or a conventional intra coding mode is selected [9].

B. View Synthesis Optimization

The estimation of a synthesized view distortion (SVD) depending on distorted depth data has been studied recently [7], [15]. We extended these SVD methods in the depth data encoding process towards a block-based method that considers the exact synthesized view distortion change (SVDC) for each block as shown in Fig. 2 [19].

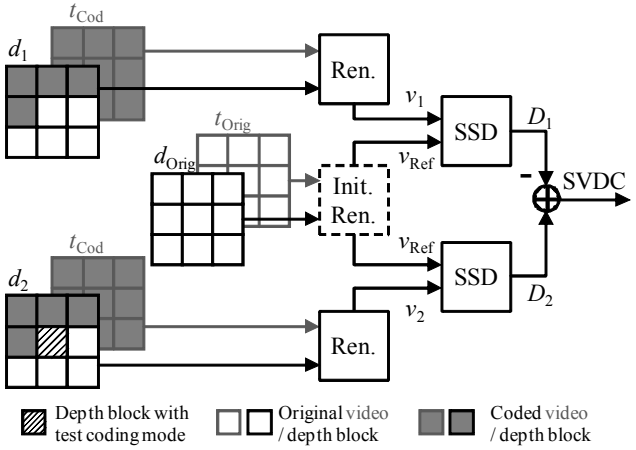

Fig. 2 Synthesized view distortion change (SVDC) calculation with respect to a currently tested depth coding mode (Ren.: view synthesis with encoder-side rendering module per block, Init.Ren.: Initial reference view synthesis per picture).

In this view synthesis optimization method, a block-wise processing aligned with depth data coding is introduced in order to provide a fast encoder operation. For an encoding optimization for one synthesized view, the method operates as follows: First, the uncoded synthesized view v_{Ref} is rendered from uncoded texture t_{Orig} and depth data d_{Orig} once per frame prior to encoding. Then, the synthesized reconstructed view is rendered separately for each block form decoded texture t_{Cod} and two variants of partially decoded/original depth data d_1 and d_2, as shown in Fig. 2. In variant 1, an original depth

block is used for rendering the reconstructed portion v_1. In variant 2, all possible coding modes are tested by using the resulting decoded depth block for rendering the reconstructed portion v_2. For both variants, the sum of squared differences (SSD) is calculated: $D_1 = \mathrm{SSD}(v_1, v_{\mathrm{Ref}})$ and $D_2 = \mathrm{SSD}(v_2, v_{\mathrm{Ref}})$. Finally, the synthesized view distortion change is obtained: $\mathrm{SVDC} = D_2 - D_1$. Thus, the optimal rendering result is obtained for a coded texture block t_{Cod}.

In our method, also parts of neighboring blocks are considered in the block-wise synthesis, that influence the rendering result with the current depth block under encoding jointly. Examples are neighboring blocks of foreground objects in original views that occlude the current block in the synthesized view. If more synthesized views are considered, the distortion measures of each single view are averaged.

III. Experimental Results

For the evaluation, the above described methods and tools have been implemented in an MVD coding framework based on the working model of HEVC, namely HM version 3.1. Here, every video and every depth sequence is coded by this algorithm. The coder configuration mostly follows the one defined in the MPEG Call for Proposals on 3D Video Coding Technology [5], testing all eight sequences with the 2-view configuration. Analyzing the effects of the new depth coding methods for the described MVD coding framework (HEVC3D) is not as straight forward as for normal video coding. The specific property that depth coding artifacts are only indirectly perceivable in synthesized video views requires analyzing the resulting quality in these views instead of the depth map directly. Consequently, the PSNR is measured for the synthesized view rendered from the 2-view video and depth data at the most critical position, i.e. the center between two original views. For rate-distortion analysis, these PSNR values are compared against the total bit rate of the compressed MVD representation.

First, the results for all test sequences are summarized in Table I, using the BD-PSNR measure [2] for calculating the percentage difference in bit rate for equal PSNR over four tested R-D points.

Table I
BJØNTEGAARD DELTA (BD) RATES FOR PROPOSED DEPTH CODING METHODS *VSO* AND *DMM*, EVALUATING OVERALL CODING PERFORMANCE.

BD results 2-view	**VSO performance**		**DMM performance**	
	DMM off BD-rate [%]	***DMM on*** BD-rate [%]	***VSO off*** BD-rate [%]	***VSO on*** BD-rate [%]
Poznan_Hall	-24.14	-23.02	-3.83	-2.59
Poznan_Street	-14.90	-14.22	-3.47	-2.61
Undo_Dancer	-32.22	-17.54	-22.47	-6.53
GT_Fly	-19.04	-14.72	-8.98	-4.20
Kendo	-7.50	-8.66	-0.46	-1.76
Balloons	-7.95	-8.06	-0.46	-0.58
Newspaper	-13.46	-13.95	-2.16	-2.75
Average	**-17.03**	**-14.31**	**-5.98**	**-3.00**

For a detailed analysis, the bit rate gains are given separately for the two coding tools: In the first and second column of Table I, the VSO performance is analyzed. The first column shows an average of 17% BD-rate gain for the case of HEVC3D with VSO versus HEVC3D without VSO. Here, DMM is disabled for both versions. The second column shows a BD-rate gain of 14% for the case, if DMM is enabled. Thus, our VSO tool achieves coding gains for both versions. In the third and fourth column of Table I, the individual DMM performance is given. Again, the BD-rate gains are shown for the HEVC3D version with DMMs versus a version without DMM. Here, coding gains of 6% are achieved for the DMM comparison without VSO and 3% for the DMM comparison with VSO. Therefore, our DMM coding method also achieves coding gains for both versions. Note, that these gains are achieved by improved depth coding methods only, while keeping the video coding identical. Furthermore, the coding gains are related to the total rate, even though the depth rate is only 10% of the total rate. For DMM, the individual gains strongly depend on the properties of the depth data. Inter-component prediction for instance requires that object edges in the depth and video component are well aligned. Therefore the bit rate savings for the two synthetic sequences *Undo_Dancer* and *GT_Fly* are considerably higher than for other (natural) sequences with estimated depth data.

For additional graphical comparison, two results from Table I are plotted as synthesized view PSNR against the total bit rate in Fig. 3.

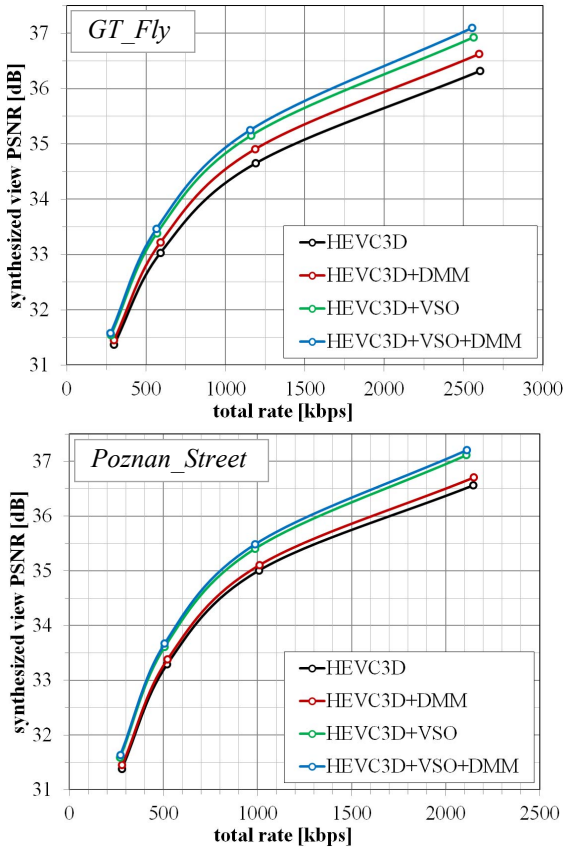

Fig. 3 R-D diagrams of GT_Fly (top) and Poznan_Street sequence (bottom), using the HEVC 3D video extension (HEVC3D) with proposed depth modeling modes (DMM) and view synthesis optimization (VSO).

The PSNR curves are shown for the synthetic *GT_Fly* sequence in Fig. 3 top, as well as for the natural *Poznan_Street* sequence in Fig. 3 bottom. Again, both sets of PSNR curves show, that a significant coding gain is achieved by enabling view synthesis optimization in the codec. Furthermore, additional coding gains are achieved with the new depth modeling modes for both cases, i.e. against the reference HEVC3D and HEVC3D with VSO. Overall, the combination of DMM and VSO gives the best coding performance with HEVC3D. Fig. 3 also shows the higher coding gains for DMM for the synthetic *GT_Fly* sequence in comparison to the natural *Poznan_Street* sequence.

IV. Conclusions

We presented advanced coding methods for 3D video compression with depth modeling modes (DMM) and encoder-side view synthesis optimization (VSO). With DMM, we proposed new and optimized algorithms for depth data coding. These methods are specifically adapted to the characteristics of depth maps, allowing for a close approximation of a depth block by non-rectangular partitions. For signaling the separation line or contour within such blocks, different methods were described, including explicit signaling, derivation from neighboring blocks and inter-component prediction from the video data. With VSO, we implemented a depth encoder optimization, which uses the exact synthesized view distortion change. This distortion measure operates block-wise and selects the best depth coding mode by considering the resulting distortion in a synthesized view. Thus, parts of the depth data belonging to occluded information in synthesized views, are omitted during depth encoding.

Our approach has been implemented as an extension to an HEVC-based 3D video codec, namely as an additional set of depth intra coding modes and Lagrange optimization for depth encoding with respect to the synthesized view distortion. The obtained experimental results showed that significant coding gains were achieved individually by each tool, even though the depth rate only contributes 10% to the overall MVD rate. Finally, the combination of DMM and VSO provided the highest coding gains, as also shown by the PSNR curves of two examples from the 3D video test set.

References

[1] P. Benzie *et al.*, “A Survey of 3DTV Displays: Techniques and Technologies”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1647-1658, Nov. 2007.

[2] G. Bjontegard, "Calculation of Average PSNR Differences between RD curves," ITU-T Q.6/SG16, doc. VCEG-M33, Austin, TX, USA, April 2001.

[3] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, “The Emerging MVC Standard for 3D Video Services,” *EURASIP Journal on Advances in Signal Processing*, Vol. 2009, No. 1, January 2009.

[4] R. Ferreira, E. Hung, R. de Queiroz, and D. Mukherjee, “Efficiency Improvements for a Geometric-partition-based Video Coder,” *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[5] ISO/IEC JTC1/SC29/WG11, “Call for Proposals on 3D Video Coding Technology”, Doc. N12036, Geneva, CH, March 2011.

[6] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 10, March 2010.

[7] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, “Depth map coding with distortion estimation of rendered view,” SPIE Conference Series, vol. 7543, Jan. 2010.

[8] J. Konrad and M. Halle, “3-D Displays and Signal Processing – An Answer to 3-D Ills?”, *IEEE Signal Processing Magazine*, vol. 24, no. 6, Nov. 2007.

[9] P. Merkle, C. Bartnik, K. Müller, D. Marpe, and T. Wiegand, “3D Video: Depth Coding Based on Inter-component Prediction of Block Partitions”, *Proc. PCS 2012, Picture Coding Symposium*, Krakow, Poland, May 2012.

[10] P. Merkle *et al.*, “The Effects of Multiview Depth Video Compression on Multiview Rendering,” *Signal Processing: Image Communication*, vol. 24, is. 1+2, pp. 73-88, Jan. 2009.

[11] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, “Efficient Prediction Structures for Multiview Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461-1473, November 2007.

[12] Y. Morvan, D. Farin and P.H.N. de With, “Platelet-based coding of depth maps for the transmission of multiview images,” *Proceedings of SPIE, Stereoscopic Displays and Applications*, vol. 6055, San Jose, USA, January 2006.

[13] K. Müller, P. Merkle, and T. Wiegand, “3D Video Representation Using Depth Maps”, *Proceedings of the IEEE, Special Issue on 3D Media and Displays*, vol. 99, no. 4, pp. 643 - 656, April 2011.

[14] H. Oh and Y. Ho, "H.264-Based Depth Map Sequence Coding Using Motion Information of Corresponding Texture Video", *Proceedings of the Pacific-Rim Symposium on Image and Video Technology*, pp.898-907, Hsinchu, Taiwan, December 2006.

[15] B. T. Oh, J. Lee, and D.-S. Park, “Depth map coding based on synthesized view distortion function,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1344 –1352, Nov. 2011.

[16] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, “Depth Reconstruction Filter and Down/Up Sampling for Depth Coding in 3-D Video”, *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 747-750, Sept. 2009.

[17] H. Schwarz and T. Wiegand, “Inter-View Prediction of Motion Data in Multiview Video Coding”, *Proc. PCS 2012, Picture Coding Symposium*, Krakow, Poland, May 2012.

[18] G. J. Sullivan and J. R. Ohm, “Recent Developments in Standardization of High-Efficiency Video Coding (HEVC)”, *Proc. SPIE*, vol. 7798, Aug. 2010.

[19] G. Tech, H. Schwarz, K. Müller, and T. Wiegand, “3D Video Coding using the Synthesized View Distortion Change”, *Proc. PCS 2012, Picture Coding Symposium*, Krakow, Poland, May 2012.

[20] M. Winken, H. Schwarz, and T. Wiegand, “Motion Vector Inheritance for High Efficiency 3D Video plus Depth Coding,” *Proc. PCS 2012, Picture Coding Symposium*, Krakow, Poland, May 2012.

[21] S. Yea and A. Vetro, “View Synthesis Prediction for Multiview Video Coding”, *Signal Processing: Image Communication*, vol. 24, is. 1+2, pp. 89-100, Jan. 2009.