Decoder Side Multiplane Images using Geometry Assistance SEI for MPEG Immersive Video

Patrick Garus, Felix Henry, Thomas Maugey, Christine Guillemot

Decoder Side Multiplane Images using Geometry Assistance SEI for MPEG Immersive Video

Patrick Garus
Orange Labs
Rennes, France
patrick1.garus@orange.com

Felix Henry
Orange Labs
Rennes, France
felix.henry@orange.com

Thomas Maugey
Inria
Rennes, France
thomas.maugey@inria.fr

Christine Guillemot
Inria
Rennes, France
christine.guillemot@inria.fr

***Abstract*—The MPEG Immersive Video (MIV) standard enables a novel technology denoted as decoder side depth estimation (DSDE) by introducing a dedicated Geometry Absent profile. In DSDE only texture information is coded and the corresponding geometry is reconstructed on the decoder side. MIV further enables the coding of side-information useful to the geometry reconstruction, denoted as Geometry Assistance SEI message. An emerging format for immersive video are Multiplane Images, which is investigated for feasibility in coding systems due to their promising rendering quality with complex sequences. In this work, we show that MIV can be used to construct block-based Multiplane Images on the decoder-side and to enhance the view synthesis performance utilizing the Geometry Assistance SEI. In a complexity-aware setting using only 32 planes, up to 6 dB of quality improvement is achieved compared to the reference.**

***Index Terms*—MPEG Immersive Video, Multiplane Images, Decoder-Side Depth Estimation**

I. Introduction

MPEG is releasing a novel standard for coding immersive video denoted as MPEG Immersive Video (MIV) or ISO/IEC MPEG-I Part 12 [1]. MIV is following a different design philosophy as opposed to the previous immersive video coding standards developed by MPEG namely the 3D extension of HEVC [2] and AVC [3]. The MIV specification does not extend a specific 2D video standard, but instead remains agnostic to the 2D video codec that is used. Consequently, it has been developed under the assumption that the geometry component of the immersive video sequence cannot benefit from dedicated depth coding tools. The main profile of the MIV standard is based on the multiview video plus depth (MVD) format, which has been traditionally used by MPEG to represent immersive video [4]. The final step of an immersive video coding system is view synthesis, which renders a novel view requested by the client. View synthesis algorithms utilizing the MVD format are often referred to as Depth Image-Based Rendering (DIBR). Popular algorithms developed in scope of the standardization activities are the View Synthesis Reference Software (VSRS) [5], Reference View Synthesizer (RVS) [6] as well as Versatile View Synthesis (VVS) [7]. They are based on a variation of projections, depth processing, view blending and hole filling (or inpainting). Nevertheless, these tools cannot overcome the inherent disadvantages of the MVD representation related to handling of advanced sequences containing non-lambertian surfaces and complex structures. Extensions of DIBR algorithms aiming to handle these sequences involve the replacement of depth maps with advanced representations [8]. Simultaneously, novel immersive video formats are designed outside of the compression domain for high-quality rendering of sequences with complex properties [9]. Recently, neural networks are utilized to learn [10] or to output a representation [12]. However, they are typically developed without the consideration of compression and may therefore be unfeasible for consideration for immersive video standards. Nevertheless, an extended profile of the MIV standard enables the coding of Multi-Plane Images (MPI). While in principle possible to be used, the MPI format has not been able to outperform the MVD format following the common test conditions (CTC) of MIV [15]. The key problem of the MPI format is the requirement of high pixel rates, which drastically reduces the view synthesis performance. A third profile of MIV is denoted as the Geometry Absent (GA) profile [18], which enables Decoder-Side Depth Estimation (DSDE). DSDE has been recently introduced as an alternative system, which overcomes many downsides of MIV, especially related to pixel rate, view synthesis quality and bitrate requirements [16].

In this paper, we propose to construct MPIs on the decoder-side utilizing the GA Profile of MIV. Being on the decoder-side, complexity is a crucial limitation of rendering algorithms. Consequently, we propose to utilize the MPI construction in a low-complexity scenario using only a limited number of planes. In order to preserve most of the quality, we perform the MPI construction on a block-basis and utilize the Geometry Assistance SEI (GA SEI) message.

The remainder of the paper is structured as follows: in chapter II, we present the fundamentals of the MIV standard, DSDE as well as MPIs, in chapter III we present our proposed block-based decoder-side MPI approach with GA SEI, chapter IV discusses the results and chapter V concludes the paper.

II. Fundamentals

In the following, we summarize the key aspects of the MIV Standard, the DSDE system as well as MPIs to set the foundation for our proposal in section III.

A. MPEG Immersive Video

The MIV standard enables 6 degrees of freedom (6DoF) by specifying a bitstream containing relevant information to

interpret the texture as well as depth atlases and to perform view synthesis [1]. It does not describe in what manner these atlases are coded, *i.e.* which 2D codec and what settings are used. Consequently, it cannot be expected that the atlases are coded with dedicated atlas tools and that the depth atlases are coded with dedicated depth coding tools. The responsibility to add support for appropriate compression of texture and depth atlases therefore lies in the 2D video standardization groups. Given that the future of depth coding is uncertain, *i.e.* due to the lack of deployment of 3D-HEVC, DSDE has emerged as an alternative system to depth coding.

B. Decoder Side Depth Estimation

DSDE is a novel system design, which has been developed in scope of the standardization process of MIV [18]. The motivation for performing the depth estimation on the decoder-side is the improved coding gain in terms of bit rate savings as well as view synthesis quality. Furthermore, depth maps contribute to the overall pixel rate requirements of the 2D codec, as they are coded as luma pixels. Pixel rate is defined as the number of luma pixels per second that can be processed by a hardware decoder, which is set by the CTC to 32 Megapixels at 30 frames per second, motivated by decoding capabilities of modern smartphones [21]. A typical test sequence with a resolution of 2048 × 1088 and 16 views requires 64 Megapixels at 30 frames per second and therefore, only 8 out of 16 views can be coded into the atlases. In case of DSDE, all 16 textures can be encoded, as the depth maps are omitted, which contribute to the pixel rate in same amounts.

The studies conducted in the context of DSDE [17] has brought an useful technology into the MIV standard denoted as the Geometry Assistance SEI (GA SEI) message [18]. The GA SEI includes the coding of side-information on a block-basis, which can be used by a depth estimator to compute depth maps with lower complexity and with higher precision. An important component of the GA SEI is locally optimal depth ranges, which we will utilize in our proposal to enhance the performance of the MPI construction. Additional syntax elements include the signaling of the partitioning as well as a skip flag for depth estimation, which are less relevant in our proposal.

C. Multiplane Images

It is important to understand that in the context of MIV, any additional attribute is coded through a 2D codec and therefore increases the need for pixel rate. For MPIs the depth map is replaced by an alpha volume, where each alpha layer refers to a fixed depth value. Therefore, the amount of data in terms of luma pixels is significantly higher for MPIs than for MVD. Even though compact representations are being studied [13] [14] the pixel rate bottleneck severely impacts the coding performance for MPIs [15]. Given the high potential of MPIs in terms of view synthesis performance [11] [12] it is desirable to make their usage feasible in compression. As only texture information is required to compute an MPI, similarly for depth maps in DSDE, it appears natural to study the MPI construction on the decoder-side effectively avoiding the pixel rate bottleneck and inefficient compression using generic 2D video codecs. It has been previously shown that MPIs can be refined utilizing depth maps [19]. However, estimating depth maps prior to the MPI construction is costly in terms of runtime and sending depth maps is costly in terms of bitrate. In the following, we will utilize the optimal depth ranges provided by the GA SEI message to construct block-based MPIs with only 32 planes enabling high-quality synthesis in a complexity-aware setup.

III. Proposed Block-based Decoder Side MPI with Geometry Assistance SEI

We present three systems in this work shown in Fig. 1: the anchor, the block-based MPI and the block-based MPI with GA SEI message.

A. Anchor

We use the test model of MIV (TMIV) 8.0 and the HEVC test model (HM) 16.23. TMIV is operated in Geometry Absent. The settings are defined by the common test conditions (CTC) of MIV [20]. The CTC does not define an MPI anchor and we are therefore required to define it ourselves. We utilize the MPI construction network of the Stereo Magnification approach [11]. This network has been trained to estimate 32 planes and is designed for view extrapolation. Given a stereo MPI approach, only two views (T_0 and T_1) need to be coded by the TMIV Encoder. These two views are packed into a texture atlas T_{ATL} and coded by HM. On the decoder-side, the atlas is unpacked and the decoded textures are provided to the MPI construction. An MPI is constructed for view T_0^*. In order to evaluate the performance, S_1^* is synthesized and compared to the uncoded reference view T_1 in terms of PSNR.

B. Block-based MPI construction

In order to take advantage of the GA SEI message, the MPI construction needs to be applied on a block-basis. The content is separated into blocks of size 128×128, according to the CTC. In order to avoid boundary effects, it is further padded by 32 pixels on the left, top and bottom boundaries and by 224 on the right boundary, see Fig. 2. These numbers have been derived empirically and are motivated by the patch-based synthesis, described below. Given a full HD frame, 135 MPI-patches are estimated independently. The view synthesis is performed on a MPI-patch as well. Each MPI-patch is projected to the target view and the padded area is removed. The process is shown in Fig. 3. Because the input of the MPI network are two patches, each synthesized patch will end up with an area that could not be recovered (grey area) and possibly an area with strong artifacts, reflecting the uncertainties in the MPI-patch. This situation motivated the strong padding on the right boundary. The padded synthesized patch is cropped to the final, synthesized patch. This process is repeated for all MPI-patches. The cropped, synthesized patches are stitched together to create the final synthesized view, see Fig. 4. The boundary effects (due to the block based

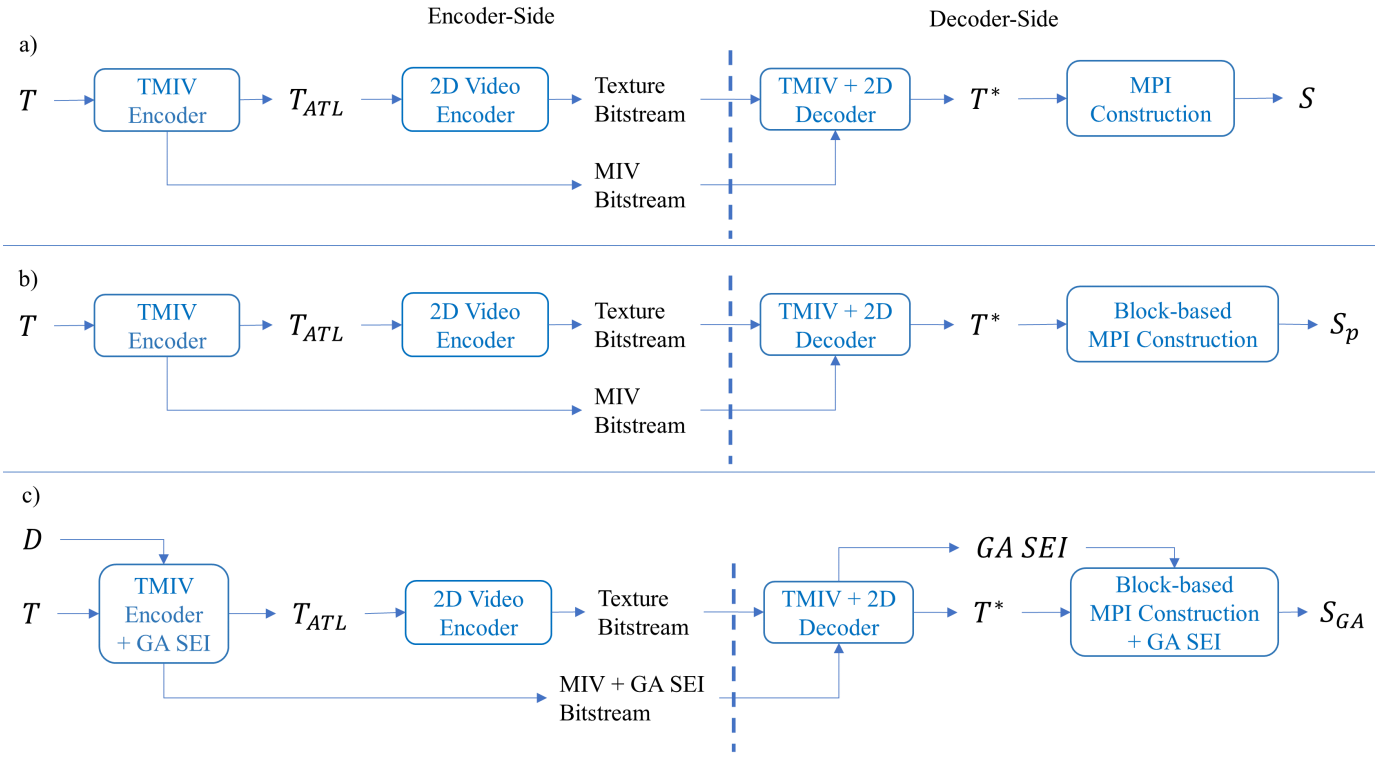

Fig. 1. The three decoder-side MPI systems discussed in this article: The unmodified system in a), the block-based system in b) and the block-based and GA-SEI enhanced system in c). Input to all systems are textures T, which are converted into Atlases T_{ATL} using the TMIV Encoder and subsequently compressed using a 2D video encoder. Depth maps D are used by the TMIV-internal feature extractor in c), in order to derive the GA SEI message. The texture bitstream is decoded by the corresponding 2D video decoder and back-converted to the multiview format T^* by the TMIV decoder. A MPI construction algorithm utilizes these textures to derive the MPIs and to synthesize novel views S, S_p and S_{GA} in systems a), b) and c) respectively. The latter utilizes additionally the GA SEI provided by the TMIV decoder.

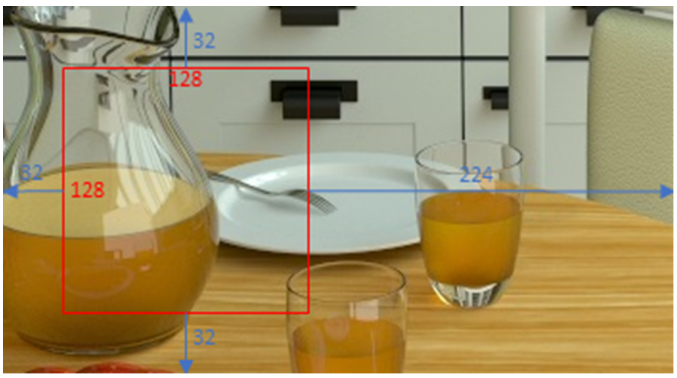

Fig. 2. Construction of the padding for a certain block (red box).

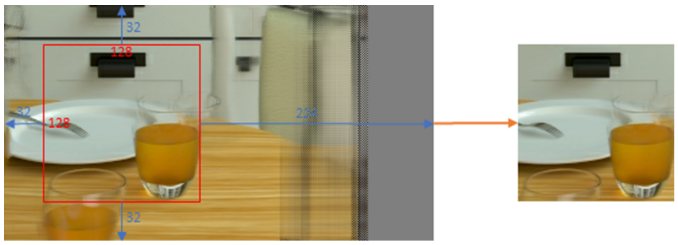

Fig. 3. Left: synthesized patch using one patch-MPI. Right: final cropped, synthesized patch.

Fig. 4. Final synthesized view, stitched together from synthesized patches.

approach) are barely visible. The main artifact remains at the far right view, which is the consequence of the chosen extrapolation approach. Due to this artifact, we restrain our PSNR evaluation on $height \times (width - 200)$, excluding this artifact in all experiments, including the anchor.

C. Enhancement through Geometry Assistance SEI

The GA SEI is derived on the encoder-side utilizing the feature extractor of TMIV. It is encoded into the MIV bitstream and therefore increases the total bitrate of the proposal,

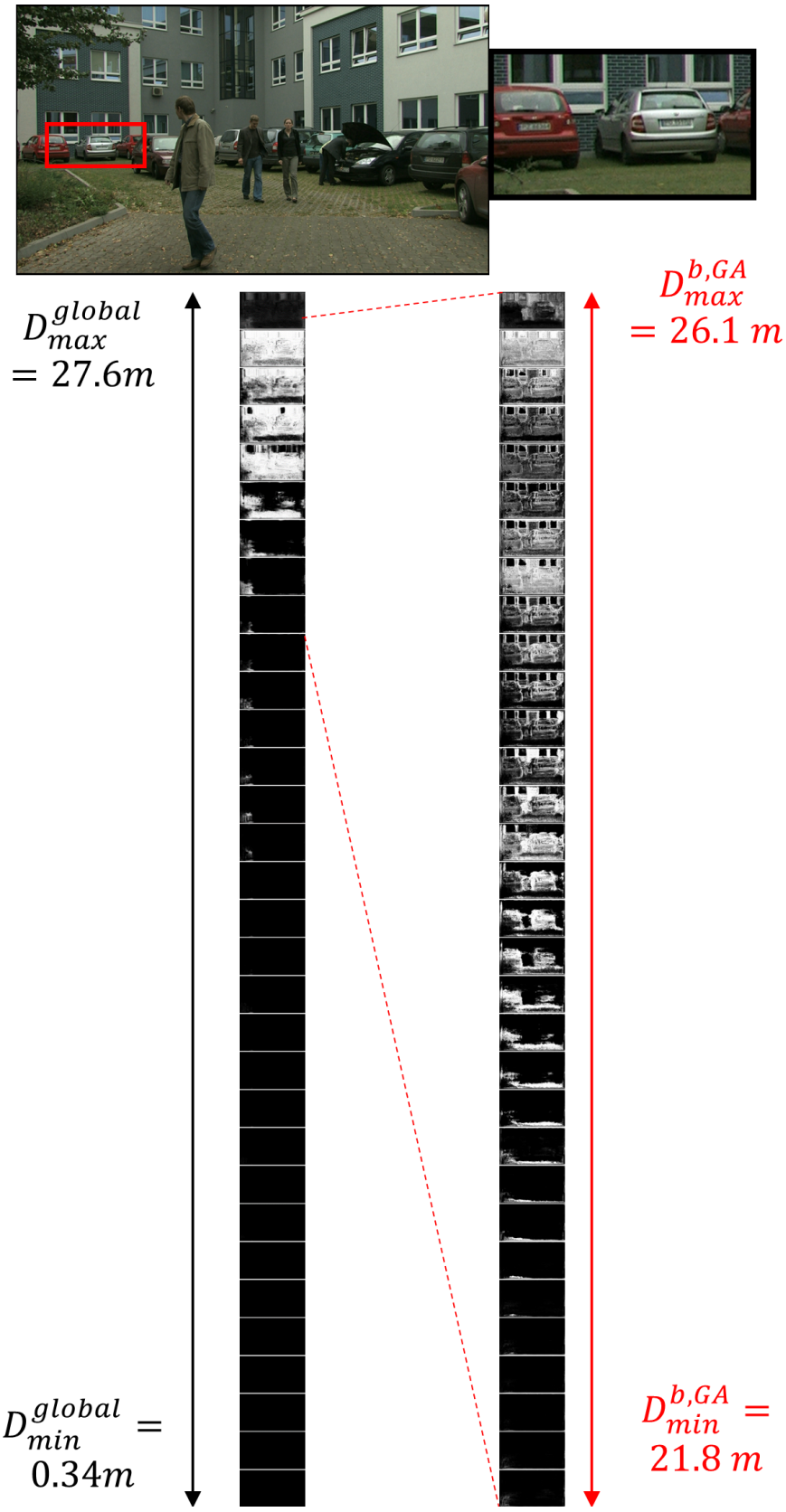

Fig. 5. Illustration of the benefit of the GA SEI. Real example based on the Carpark sequence. For the given block, the GA SEI allows to analyze the relevant depth range, leading to more meaningful transparency layers.

which is considered in the evaluation section. However, it does not affect the coded texture bitstream and therefore, all synthesis PSNR improvements are due to the GA SEI. On the decoder-side, the GA SEI is provided to the block-based MPI construction algorithm. Utilizing the GA SEI, the relevant depth range is known to the MPI constructor. The 32 planes, available by the Stereo Magnification algorithm, can therefore be placed in-between the relevant depth range in every block. Consequently, a denser sampling in the depth space can be achieved from content, which is truly relevant. In the anchor system, content outside the valid depth range, and therefore, outside the volumetric video will be sampled. This situation is demonstrated in Fig. 5, which shows the alpha maps for all 32 depth planes sampled in the reference and our proposal for a given block in a real example. We visualize a patch of the Carpark sequence, which has a global depth range of $[D_{min}^{global}, D_{max}^{global}] = [0.34\ m, 27.6\ m]$. Consequently, an MPI layer is constructed approximately every meter in the given example. It can be observed, that the majority of transparency layers are zero, because the corresponding content is not present in processed block. Zero-layers do not contribute to the alpha-blending process during view synthesis and are therefore irrelevant. In the GA SEI enhanced proposal however, the relevant depth range for this block b is identified as $[D_{min}^{b,GA}, D_{max}^{b,GA}] = [21.8\ m, 26.1\ m]$ and the available 32 depth planes are placed in this range. Consequently, the sampling is more dense in the range, in which relevant content exists and the majority of layers are non-zero. In other words, more relevant information from the volumetric video is captured in the 32 planes utilizing the GA SEI.

IV. Experimental Results

When analysing the results, we have observed, that the MPI-specific artifacts are rather constant among all QPs. Even further, we have observed a strong robustness of the CNN-based stereo magnification approach towards compressed input. Consequently, the visual results shown in the following are computed using *uncoded* input textures. Fig. 6 shows two examples of view synthesis of the anchor and the proposal utilizing the Geometry Assistance SEI message. Without increasing the number of depth planes, the resulting synthesized images become significantly sharper. This is because the depth planes have been locally placed, where the content truly exists and therefore, much more of the available 32 planes contain relevant information. If the depth range of the full frame is used, the depth planes may be locally placed without covering any existing information, as they are distributed uniformly. In the estimated MPIs, those planes are often found to be "empty", meaning, the transparencies are zero and the area does not carry meaningful information. Furthermore, we observe that by narrowing down the range, the extrapolation works better and more information can be recovered. The fan sequence is particularly interesting as it shows where the depth-driven approach may fail. In case of a "fence"-like object, which allows to see through the object, the transmitted depth ranges lose most of their benefit.

TABLE I
PSNR performance of the anchor, block-based MPI with and without Geometry Assistance SEI.

Sequence	Anchor	block-MPI (w/o GA SEI)	ΔPSNR	GA-MPI	ΔPSNR
Painter	33.58	33.37	-0.21	32.94	-0.63
Mirror	23.98	23.56	-0.41	25.83	+1.86
Kitchen	27.45	27.40	-0.05	32.18	+4.87
Frog	22.96	22.89	-0.07	24.72	+1.75
Fan	20.26	20.07	-0.18	22.82	+2.57
Carpark	28.98	28.79	-0.19	30.69	+1.70
Hall	29.25	28.92	-0.32	35.42	+6.16
Street	30.49	30.34	-0.15	34.13	+3.64
Average	27.12	26.92	-0.20	29.84	+2.74

Tab. I summarizes the synthesis PSNR for the uncoded sequences. As expected, due to the limitation to 32 planes, the anchor performance varies strongly from sequence to sequence ($20.26\ dB$ to $33.58\ dB$). The Table confirms, that the impact of moving to a block-based MPI approach is low. On average, a loss of $-0.2\ dB$ originates from the block-based computation of the MPIs. The Geometry Assistance SEI improves the quality from $1.75\ dB$ for the Carpark sequence to up to $6.16\ dB$ for the Hall sequence. The only exception is the Painter sequence, which show a loss of $-0.63\ dB$. This odd result is however consistent with previous DSDE work [16] [17] and may be related to the depth maps quality. On average, a quality improvement of $2.74\ dB$ is observed if the Geometry Assistance SEI is utilized.

V. Conclusion

In this paper, we proposed to utilize the Geometry Assistance SEI message of the MIV standard to significantly enhance the view synthesis performance given a setup of reduced complexity. With merely 32 MPI planes, the synthesis performance is increased by up to 6.12 dB and 2.74 dB on average. Furthermore, we show that recent rendering methods not relying on explicit depth maps can be utilized with the MIV standard using the Geometry Absent profile, given that depth estimation and view synthesis are not restricted by the specification. We have shown that the Geometry Assistance SEI message can have a significant benefit, which is not limited to depth estimation.

References

[1] J. M. Boyce et al., "MPEG Immersive Video Coding Standard," in Proceedings of the IEEE, vol. 109, no. 9, pp. 1521-1536, Sept. 2021, doi: 10.1109/JPROC.2021.3062590.

[2] G. Tech, Y. Chen, K. Müller, J. -R. Ohm, A. Vetro and Y. -K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 35-49, Jan. 2016, doi: 10.1109/TCSVT.2015.2477935.

[3] A. Vetro, T. Wiegand and G. J. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard," in Proceedings of the IEEE, vol. 99, no. 4, pp. 626-642, April 2011, doi: 10.1109/JPROC.2010.2098830.

[4] P. Merkle, A. Smolic, K. Muller and T. Wiegand, "Multi-View Video Plus Depth Representation and Coding," 2007 IEEE International Conference on Image Processing, 2007, pp. I - 201-I - 204, doi: 10.1109/ICIP.2007.4378926.

Fig. 6. Examples of the synthesis performance given 32 Planes for the Hall and Fan sequence.

[5] Stankiewicz, Olgierd and Wegner, Krzysztof and Tanimoto, Masayuki and Domański, Marek. (2013). Enhanced View Synthesis Reference Software (VSRS) for Free-viewpoint television.

[6] S. Fachada, D. Bonatto, A. Schenkel and G. Lafruit, "Free Navigation in Natural Scenery With DIBR: RVS and VSRS in MPEG-I Standardization," 2018 International Conference on 3D Immersion (IC3D), 2018, pp. 1-6, doi: 10.1109/IC3D.2018.8657912.

[7] Joël Jung, Patrick Boissonade. VVS: Versatile View Synthesizer for 6-DoF Immersive Video. 2020. ffhal-02541110f

[8] S. Fachada, D. Bonatto, Y. Xie, P. R. Alface, M. Teratani and G. Lafruit, "Depth Image-Based Rendering of Non-Lambertian Content in MPEG Immersive Video," 2021 International Conference on 3D Immersion (IC3D), 2021, pp. 1-6, doi: 10.1109/IC3D53758.2021.9687263.

[9] Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. ACM Trans. Graph. 36, 6, Article 235 (December 2017), 11 pages. https://doi.org/10.1145/3130800.3130855

[10] Wang, Qianqian and Wang, Zhicheng and Genova, Kyle and Srinivasan, Pratul and Zhou, Howard and Barron, Jonathan and Martin Brualla, Ricardo and Snavely, Noah and Funkhouser, Thomas. (2021). IBRNet: Learning Multi-View Image-Based Rendering.

[11] Zhou, Tinghui and Tucker, Richard and Flynn, John and Fyffe, Graham and Snavely, Noah, "Stereo Magnification: Learning View Synthesis using Multiplane Images", SIGGRAPH, 2018.

[12] Mildenhall, Ben and Srinivasan, Pratul P. and Ortiz-Cayon, Rodrigo and Kalantari, Nima Khademi and Ramamoorthi, Ravi and Ng, Ren and Kar, Abhishek., "Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines"

[13] J. Navarro and N. Sabater, "Compact And Adaptive Multiplane Images For View Synthesis," 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 3403-3407, doi: 10.1109/ICIP42928.2021.9506403.

[14] B. Vandame et al., "Pipeline for Real-Time Video View Synthesis," 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2020, pp. 1-6, doi: 10.1109/ICMEW46912.2020.9105988.

[15] J. Fleureau and R. Doré and B. Chupeau and F. Thudor and G. Briand and T. Tapie, "MIV CE1-Related - Activation of a transparency attribute and application to MPI encoding", ISO/IEC JTC 1/SC 29/WG 04 MPEG/m55089, Oct. 2020.

[16] P. Garus, J. Jung, T. Maugey and C. Guillemot, "Bypassing Depth Maps Transmission For Immersive Video Coding," 2019 Picture Coding Symposium (PCS), 2019, pp. 1-5, doi: 10.1109/PCS48520.2019.8954543.

[17] P. Garus, F. Henry, J. Jung, T. Maugey and C. Guillemot, "Immersive Video Coding: Should Geometry Information Be Transmitted as Depth Maps?," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 5, pp. 3250-3264, May 2022, doi: 10.1109/TCSVT.2021.3100006.

[18] D. Mieloch et al., "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," in IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2022.3162916.

[19] A. Li, L. Fang, L. Ye, W. Zhong, and Q. Zhang, "Geometry-guided view synthesis with local nonuniform plane-sweep," in Digital TV and Wireless Multimedia Communication (Communications in Computer and Information Science), vol. 1181, G. Zhai, J. Zhou, H. Yang, P. An, and X. Yang, Eds. Singapore: Springer. 2020.

[20] "Common Test Conditions for MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0085, Online, April 2021.

[21] B. Kroon, V. K. M. Vadakital, and J. Jung, "Recommended Pixel Rate Limits for the CTC for Immersive Video", ISO/IEC JTC1/SC29/WG11/MPEG2019/M49826, Jul. 2019.