Bypassing Depth Maps Transmission For Immersive Video Coding

Patrick Garus, Jung Joel, Thomas Maugey, Christine Guillemot

Bypassing Depth Maps Transmission For Immersive Video Coding

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

Joel Jung
Orange Labs
Guyancourt, France
joelb.jung@orange.com

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

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

***Abstract*—This paper addresses several downsides of the system under development in MPEG-I for coding and transmission of immersive media. We present a solution, which enables Depth-Image-Based Rendering for immersive video applications, while lifting the requirement of transmitting depth information. Instead, we estimate the depth information on the client-side from the transmitted views. The approach leads to an impressive rate saving (37.3% in average). Preserving perceptual quality in terms of MS-SSIM of synthesized views, it yields to 24.6% rate reduction for the same quality of reconstructed views after residue transmission under the MPEG-I common test conditions. Simultaneously, the required pixel rate, *i.e.* the number of pixels processed per second by the decoder, is reduced by 50% for any test sequence. To the author's knowledge, this is the first time that such an approach is under consideration in the context of immersive video coding.**

***Index Terms*—MPEG, immersive video coding, depth estimation**

I. Introduction

The ISO/IEC motion pictures experts group (MPEG) has been for several years investigating solutions for delivering immersive computer generated or captured video content. A work item referred to as ISO/IEC 23090 and known as MPEG-I has been started in 2017. Current state-of-the-art technologies for immersive video coding can be categorized according to the degrees of freedom (DoF) of positioning and orientation of the viewer [1], [2]. Enabling merely yaw, pitch and roll movements of the users head provides the viewer a total of *3DoF*. The first specification finalized by MPEG-I Visual for 3DoF is referred to as Omnidirectional MediA Format (OMAF). The next step has been in allowing small translational movements of the head, leading to a *3DoF+* standard also referred to as Metadata for Immersive Video (MIV) that is expected to be finalized in October 2019. Current standardization activities are targeting the specification of a solution that would enable the user to freely navigate through the content thanks to three additional translations of the body, *i.e.* with six degrees of freedom (*6DoF*). Extending 3DoF+ (or the MIV solution) towards 6DoF is a challenging task with several open questions. MIV has been designed for short baseline scenarios and little user's motion. It further establishes a strong dependency between texture compression efficiency and depth maps quality. However, estimating high quality depth maps is challenging in the 6DoF context and the baselines are much larger. In addition, depth maps estimated from uncompressed views may not be the most appropriate for view synthesis in presence of quantization noise. The compression of the depth maps may also impact the quality of the synthesized views. This makes it difficult to select the right encoder configuration for the depth maps, *e.g.* choosing the right depth quantization parameter QP_D depending on the QP_T used for texture coding. Furthermore, depending on the level of details in the estimated depth maps, their transmission can represent up to 30% of the total bitstream. It is also desirable to reduce the pixel rate, *i.e.* the number of pixels processed per second by the decoder.

In this paper, we propose an architecture addressing the above challenges and drawbacks of the 6DoF solution under consideration in MPEG-I Visual. The proposed approach yields an average Bjøntegaard delta (BD) rate [6] of 37.3%. Simultaneously, the perceptual quality of synthesized views is preserved in terms of MS-SSIM compared to the MV-HEVC anchor defined in the MPEG-I Visual common test conditions [16]. The remainder of the paper is organized as follows. Section II gives a brief overview of main depth estimation and view synthesis techniques including the MPEG-I Visual reference methods. Section III describes the proposed 6DoF solution. The experimental results are presented in Section IV and Section V concludes the paper.

II. Immersive Video Coding

In contrast to 2D video, immersive applications require a richer sampling of the light rays emitted by the surrounding environment. A sparse sampling with large baselines can be achieved using several calibrated omnidirectional or conventional 2D cameras positioned at different world coordinates. Their relative positions and orientations are given by the camera parameters. One can reconstruct a denser representation of the continuous light-field by interpolating intermediate views using view synthesis algorithms *e.g.* relying on depth image-based rendering (DIBR) techniques. Depth can be estimated from the available views. However, choosing the best depth estimator for a view synthesis task is not trivial and even for CGI content, ground truth depth maps may not yield the best synthesized views [7]. Having depth maps available, the corresponding synthesizer can make use of DIBR techniques. In summary, three components distinguish immersive applica-

tions from 2D video: compression and transmission of multiple viewpoints, view synthesis and depth estimation. Since a large quantity of work has been done in the domains of depth estimation and view synthesis, only the most recent algorithms are presented in following sections before introducing the MPEG-I Visual framework.

A. Multiview Video Coding

In order to remove the redundancy between different camera views, Multiview HEVC (MV-HEVC) has been developed as an extension to HEVC [13]. It is able to efficiently remove inter-view redundancy by considering information from previously encoded views with only high-level syntax changes compared to HEVC. Having DIBR in mind, 3D-HEVC additionally considers inter-component redundancy between depth maps and textures. In this context, a lot of research has been done in improving compression of depth maps for the goal of view synthesis [5]. This is done by modelling the distortion of a synthesized view according to changes in the depth values due to the compression process [3], [4].

B. View Synthesis

A typical and recent example of a DIBR system is presented in [8], which covers 3D warping, occlusion handling by inpainting and depth map processing. Utilizing bi-directional warping, multiple virtual depth maps are generated at the target view position from up to four available views. Color-correction among views is performed and erroneous depth values are detected and filtered in the process. Inpainting is used to fill remaining occlusions in the synthesized view. Finally, edges are slightly smoothed to improve the perceptual quality. In [9], a deep-learning based approach is proposed. Depth maps are estimated using two different stereo-matching algorithms, which complement each other: one is targeting global consistency, while the other includes fine details. In a refinement step, both depth maps are fused and a mesh is created. The input to the network are the mesh and the reprojected image mosaics. Temporal consistency is encouraged in the loss term, which comprises a perceptual loss using activations at different scales of a pretrained VGG16 network.

C. Depth Estimation

The authors of [10] utilize deep learning methods to estimate depth maps for light-fields. Using a fine tuned FlowNet 2.0 network, several candidate depth maps are estimated between the target view and other horizontal or vertical views of variable distance. After normalization, the depth maps refer to the immediate neighboring view and are fused together to a single depth map. As a final step, the depth map is refined using a second CNN, which is designed as an encoder-decoder architecture. Besides of depth accuracy, the authors of [11] set the emphasis on reducing the complexity of the depth estimation algorithm by using superpixels as their basic data units. Using a GPU-optimized implementation, they achieve to estimate around one HD depth map per second. The authors of [12] use several runtime optimization strategies on their deep convolutional encoder-decoder design, such as depthwise decomposition, network pruning and hardware-specific compilation. Their CNN-based monocular depth estimator achieves up to 178 fps on 224x224 resolution video, while maintaining state-of-the art accuracy.

In this study, we focus on the algorithms used in the context of the MPEG-I Visual framework, in order to show the benefit of estimating depth at the decoder.

D. MPEG-I Framework

Fig. 1 shows the generic block diagram of the state-of-the-art 6DoF-architecture with its three main components: depth estimation, compression and synthesis. Given a set of multiview frames T, the corresponding depth maps D are first estimated using a depth estimation algorithm. Using the decoded textures T^* and depth maps D^*, intermediate views S are rendered using a synthesizer. According to the placement of the depth estimation algorithm, we denote this system as encoder-side depth estimation (ESDE).

In the current MPEG anchor, MV-HEVC is used to compress the texture and depth maps. Depth maps are estimated using DERS, which is continuously refined in the scope of MPEG and has reached version 8.0 in the 126th MPEG meeting in Geneva [14]. DERS8.0 supports up to four neighboring views to estimate a single depth map. The algorithm entails two core steps: first, different depth hypotheses are tested using block matching and the cost for each pixel and depth candidate is computed using a photo similarity measure. Second, the error cost is used to calculate the disparity for each pixel using a graph cut algorithm. For video sequences, DERS8.0 allows the usage of a tool denoted as *Temporal Enhancement*. It decreases execution time and improves temporal consistency of the estimated depth maps.

The reference synthesizer is Versatile View Synthesizer 2.0 (VVS) [15], which was designed for optimizing perceptual quality of virtual views, considering the presence of compression artifacts in the transmitted reference views. Reference views are sorted according to their warping quality, which increases its robustness towards complex camera setups. Depth maps are analyzed to avoid confusion between fore- and background objects. They are refined and textures are projected to the virtual view position. After merging, temporal inpainting is performed to fill remaining holes.

III. Proposed Framework

We introduce a modified 6DoF architecture, which bypasses the transmission of depth maps by assigning the depth estimation process to the decoder side, shown in the right side of Fig. 1. Accordingly, we denote this approach as decoder-side depth estimation (DSDE). The main difference between the suggested DSDE method and the conventional ESDE approach is the positioning of the depth estimation process, which is moved entirely to the decoder side. Nevertheless, it has extensive consequences: the first major advantage of this method is, that compression and transmission of depth maps can be bypassed entirely, hence, reducing the total bitrate.

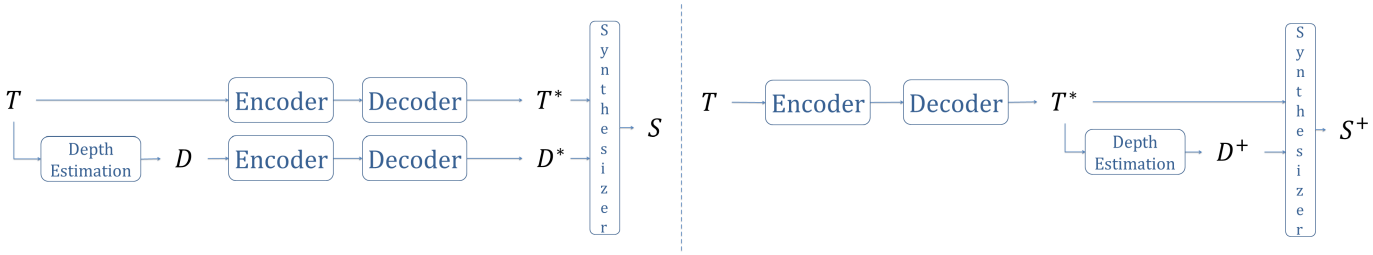


Fig. 1: left: 6DoF system used in MPEG-I Visual. Right: proposed system.

Besides of bitrate, the number of decodable pixels per second is a limiting constraint for mobile devices. The DSDE system reduces the pixel rate by 50%, making immersive applications more feasible. The second advantage is, that it is no longer necessary to find the best relation between QP_T and QP_D for depth compression. Moreover, the input signals of the depth estimation process are the decoded textures making sure it uses the same texture information as the synthesizer. The latter aspect may on the one hand introduce a different challenge for the depth estimation algorithm, due to the existence of compression artifacts: DERS has always been developed considering uncompressed textures. Neighboring decoded views will suffer from different compression artifacts, like quantization noise and blockiness, which affects the depth estimation algorithm. On the other hand, it makes sure, that the extracted features match the decoded textures instead of the source textures, as the former is used by the synthesizer. We denote the estimated depth maps in the DSDE architecture as D^+. In the following section, the synthesized textures S^+ will be compared to the anchor result S.

IV. Experimental Results

We compare our architecture to the current MPEG anchor and apply the same configurations as defined in the CTC of MPEG-I Visual [16]. The Temporal Enhancement feature of DERS leads to a huge degradation of quality over time in version 8.0 [17]. Consequently, it is not used in our setup. The test set consists of nine sequences and five QP pairs for texture and depth encoding. For objective evaluation, we compute MS-SSIM and PSNR between the synthesized view at source position and the corresponding source texture. MS-SSIM is averaged over all QPs. Low and medium bitrate ranges are analysed using the four highest and lowest QP pairs respectively. Two BD-Rates are reported, which consider Y-PSNR of either decoded textures (video) or of synthesized textures (synth). Results for all test sequences are reported in Table I.

A. Rate-Distortion performance

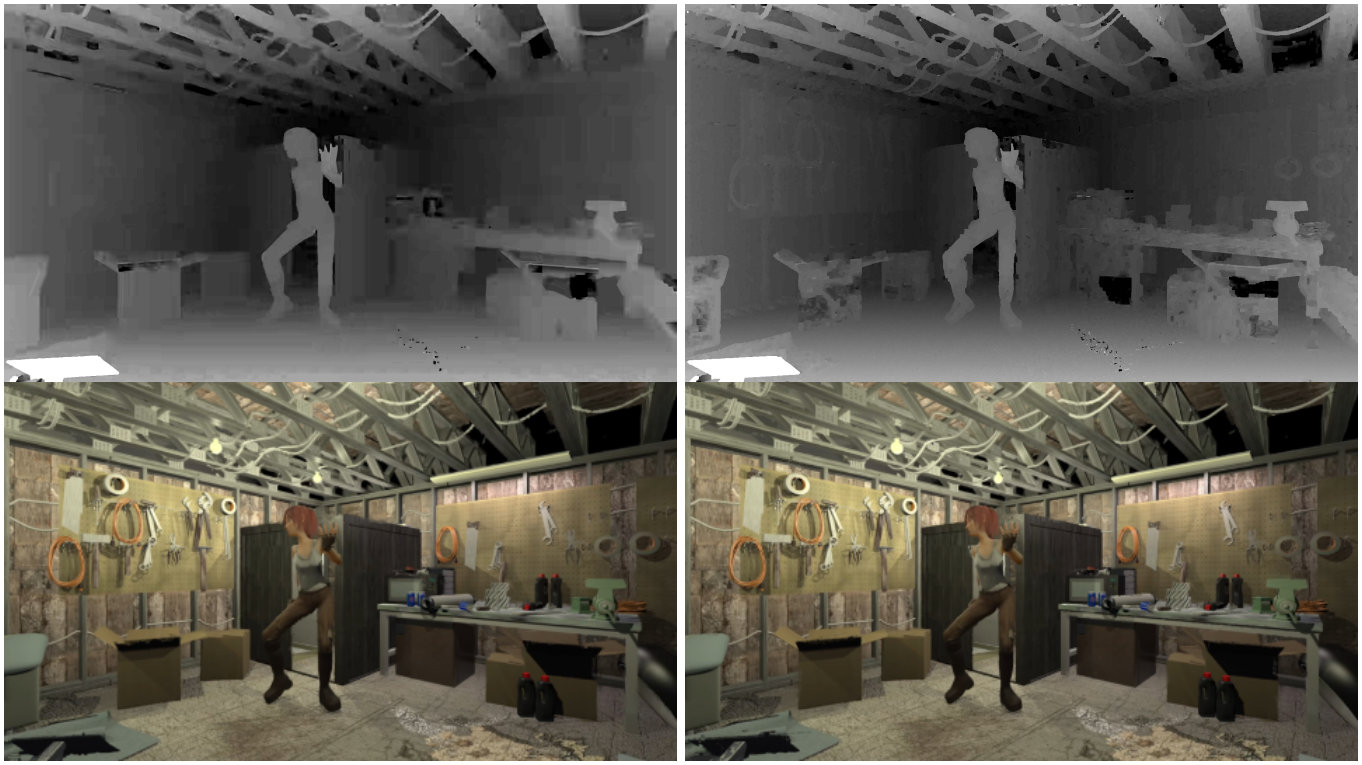
Video BD-Rate reflects the bitrate saved by the proposed system due to bypassing the depth compression. The required rate for depth compression varies due to complexity of the scenery and due to quality of the estimated depth maps. Particularly PFencing achieves the highest savings with 48.5% and 53.6% video BD-Rate savings, proving that its depth maps are difficult to compress. This is because PFencing is a natural scene, sparsely captured by cameras in an arc configuration. Containing a lot of motion and homogeneous areas, high quality depth maps are challenging to estimate. In contrast, ODancing is a CGI scene, with much denser camera spacing than PFencing. DERS estimates much clearer depth maps, which are consequently easier to compress. UUnicornA and UUnicornB form an exception as they are both multiview still images. Due to the lack of movement in the scenery, the rate required to transmit the depth maps is comparably low. Over all test sequences, the DSDE system achieves 21.2% and 24.6% average video BD-Rate savings for medium and low bitrate ranges respectively.

B. Rate-Distortion performance for synthesized views

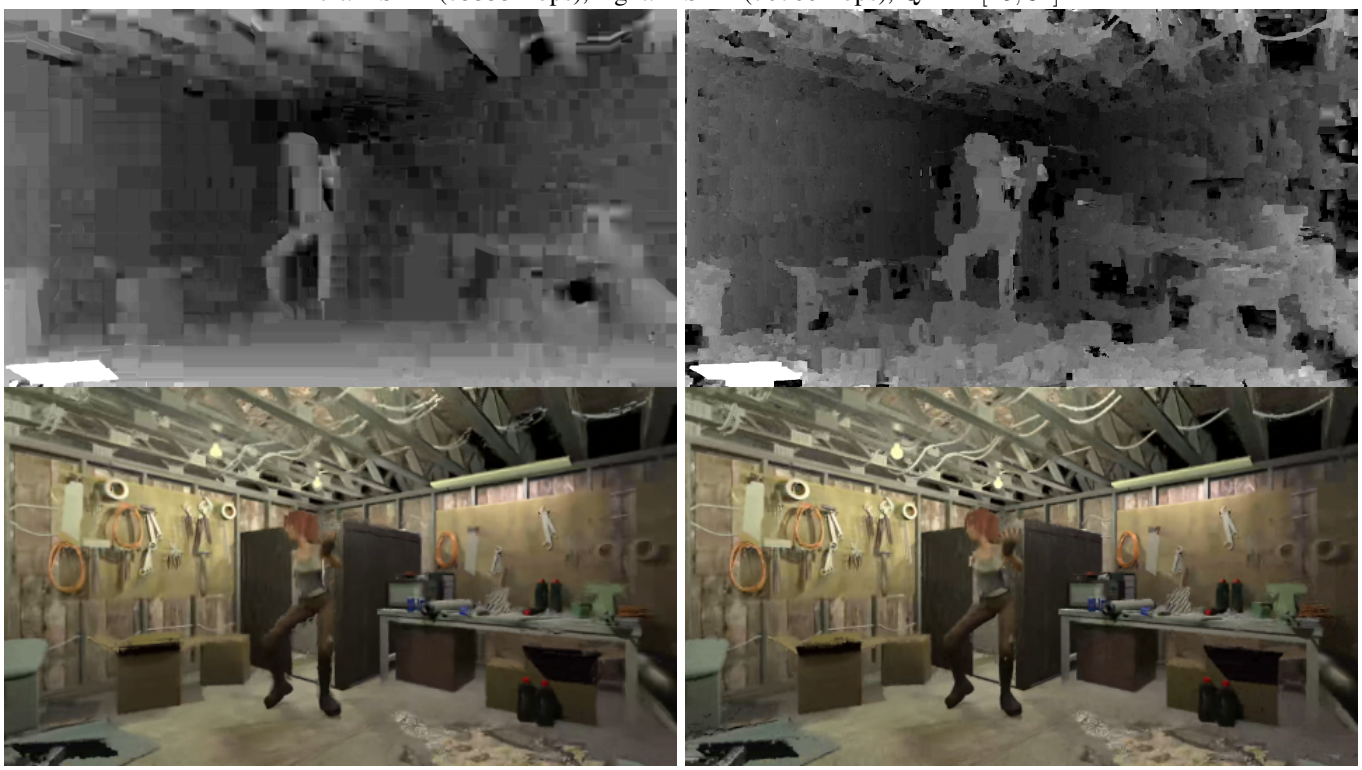
Besides of the bitrate being saved by omitting depth transmission, synth BD-Rate additionally considers the difference in PSNR of the synthesized views and therefore, the overall gain of the DSDE system. Selected corresponding rate-distortion (RD) curves are shown in Fig. 3. Synth BD-Rate for each test sequence is shown in Table I. As reflected by the synth BD-Rate, not only bitrate is saved but also synthesis quality is either maintained or improved significantly. In addition to the reasons mentioned above, the impact of additional compression artifacts are most severe for PFencing. Consequently, synth BD-Rate is lower for PFencing. Nevertheless, the compromise in rate reduction justifies the slight loss in quality. In contrast, the improvements of ODancing and IFrog are outstanding, as the quality of synthesized views increases remarkably, which shows the potential advantages of this architecture even for cases where the additional rate to spend for depth is very low compared to texture. Overall, we report average synth BD-Rate savings of 37.3% and 36.7% for medium and low bitrate ranges respectively.

C. Qualitative analysis

Fig. 2 shows depth maps and synthesized views for two QP pairs for ODancing. In the ESDE architecture, depth maps were estimated using the same source texture, but suffer from stronger compression artifacts with higher QP values. As a consequence, areas turn more homogeneous and edges loose sharpness. In comparison to the DSDE architecture, depth maps are estimated from decoded textures based on compression with different QP values. Block artifacts found in the texture become visible in the corresponding estimated depth maps. Despite its less clean appearance, depth maps originating from the DSDE architecture serve an equivalent or



Left: ESDE (73553 kbps), right: DSDE (70766 kbps), $QP = [25, 34]$



Left: ESDE (3912 kbps), right: DSDE (3310 kbps), $QP = [45, 48]$

Fig. 2: Depth maps and synthesized views for two different QP pairs of the ODancing sequence. Indicated QP pairs refer to texture and depth QP used in the ESDE architecture, *i.e* $QP = [QP_T, QP_D]$.

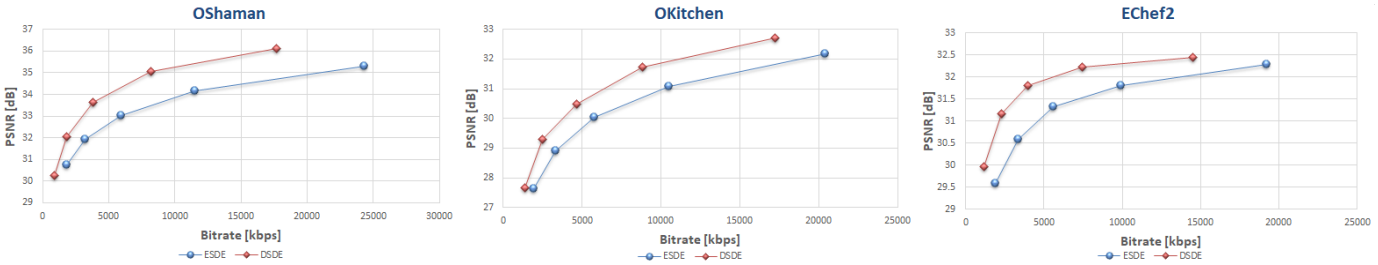


Fig. 3: Selected RD curves, comparing the DSDE to the ESDE system. BD-Rate values are reported in Tab. I.

TABLE I: BD-Rate and MS-SSIM per test sequence.

Sequence	Medium BD-Rate [%]		Low BD-Rate [%]		MS-SSIM	
	video	synth	video	synth	ESDE	DSDE
TPainter	-36.0	-31.7	-39.2	-35.1	0.9487	0.9448
UUnicornA	-4.7	6.0	-5.4	-6.7	0.9744	0.9746
UUnicornB	-5.6	-3.6	-6.7	-13.8	0.9743	0.9745
OShaman	-32.9	-55.2	-39.7	-50.0	0.9217	0.9212
OKitchen	-18.2	-39.2	-22.3	-36.0	0.9444	0.9452
ODancing	-5.3	-72.4	-8.0	-51.4	0.9749	0.9738
EChef2	-27.8	-58.4	-31.2	-54.3	0.9456	0.9478
IFrog	-11.4	-57.8	-15.0	-42.8	0.8968	0.9036
PFencing	-48.5	-23.2	-53.6	-39.9	0.9276	0.9248
Average	**-21.2**	**-37.3**	**-24.6**	**-36.7**	**0.9454**	**0.9456**

even better purpose for view synthesis compared to the ESDE depth maps. This can be seen by comparing the provided synthesized views. In the provided example, the distortion in the depth maps of the ESDE system lead to blending artifacts and blurriness in the synthesized views. In contrast, the depth maps used in the DSDE system adopt the block artifacts of the decoded textures used in DERS, which becomes more obvious for high QPs. Instead of blurriness, additional noise can be observed in textured areas. However, this circumstance does not outweigh the benefits: the depth maps used in the DSDE system and consequently the synthesized views are significantly sharper. Most of the double-contouring is avoided compared to the ESDE system. The benefit of the DSDE system is further reflected quantitatively in the MS-SSIM values, which overall proves that both systems perform similar in terms of perceptual quality, while bitrate is reduced by 21.2% and 24.6%.

V. Conclusion

In this paper, an effective system for 6DoF applications is introduced. Without any additional optimization effort, the new approach leads to impressive BD-Rate improvements for the majority of the test sequences: 37.3% average and up to 72.4% peak gain. Perceptual quality is retained according to similar MS-SSIM. The DSDE system does not rely on finding an optimal QP_D, simplifying encoding significantly. Removing depth maps from the decoder task reduces pixel rate by 50%. We are optimistic that DERS can be replaced by real-time depth estimation techniques. We are convinced that this system is laying a foundation for a series of studies, that seek to improve the way view synthesis for immersive application is performed.

References

[1] M.-L. Champel, R. Koenen, G. Lafruit, M. Budagavi, "Proposed Draft 1.0 of TR: Technical Report on Architectures for Immersive Media," ISO/IEC JTC1/SC29/WG11/MPEG2018/N17865, April 2018, San Diego, US.

[2] M. Wien, J.M. Boyce, T. Stockhammer, W.-H. Peng, "Standardization Status of Immersive Video Coding," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 5-17, March 2019.

[3] W. Kim, A. Ortega, P. Lai, D. Tian, "Depth Map Coding Optimization Using Rendered View Distortion for 3D Video Coding," in IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 3534-3545, November 2015.

[4] B. Rajei, T. Maugey, P. Frossard, "Rate-distortion analysis of multiview coding in a DIBR framework," annals of telecommunications - annales des télécommunications. 68. 10.1007/s12243-013-0375-6, November 2012.

[5] C. Debono, M. Domański, S. De Faria, K. Klimaszewski, L. Lucas, N. Rodrigues, K. Wegner, "Efficient Depth-Based Coding," in 3D Visual Content Creation, Coding and Delivery, pp.97-114, January 2019.

[6] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T Q.6/16, Doc. VCEG-M33, March 2001.

[7] A. Q. de Oliveira, T. L. T. da Silveira, M. Walter, C.R. Jung, "On the Performance of DIBR Methods When Using Depth Maps from State-of-the-art Stereo Matching Algorithms," in IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, United Kingdom, 2019, pp. 2272-2276.

[8] B. Ceulemans, S. Lu, G. Lafruit, A. Munteanu, "Robust Multiview Synthesis for Wide-Baseline Camera Arrays," in IEEE Transactions on Multimedia, vol. 20, no. 9, pp. 2235-2248, Sept. 2018.

[9] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, G. Brostowruit, "Deep Blending for Free-Viewpoint Image-Based Rendering," ACM Trans. Graph. 37 (2018): 257:1-257:15.

[10] J. Shi, X. Jiang, C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," IEEE International Conference on Acoustics, Speech, and Signal Processing, 13-17 May 2019.

[11] A. Chuchvara, A. Barsi, A. Gotchev, "Fast and Accurate Depth Estimation from Sparse Light Fields," arXIV preprint arXiv:1812.06856, December 2018.

[12] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, V. Sze, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," IEEE International Conference on Robotics and Automation, March 2019.

[13] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, Y.-K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, January 2019;26(1):35-49.

[14] T. Senoh, N. Tetsutani, H. Yasuda and M. Teratani, "Revised Proposed Depth Estimation Reference Software (pDERS8.1)," ISO/IEC JTC1/SC29/WG11/MPEG2018/m45265.v3, January 2019, Marrakesh, Morocco.

[15] P. Boissonade and J. Jung, "[MPEG-I Visual] Improvement of VVS1.0.1," ISO/IEC JTC1/SC29/WG11/MPEG2019/m46263, January 2019, Marrakesh, Morocco.

[16] J. Jung, B. Kroon, R. Doré, G. Lafruit and J. Boyce, "CTC on 3DoF+ and Windowed 6DoF (v2)," ISO/IEC JTC1/SC29/WG11/MPEG2018/N17726, July 2018, Ljubljana, Slovenia.

[17] P. Garus, J. Jung, P. Boissonade, "[MPEG-I Visual] Evaluation of DERS modifications proposed in [m45265]," ISO/IEC JTC1/SC29/WG11/MPEG2019/m46799, March 2019, Geneva, Switzerland.