PATCH DECODER-SIDE DEPTH ESTIMATION IN MPEG IMMERSIVE VIDEO

Marta Milovanovic, Felix Henry, Marco Cagnazzo, Joel Jung

PATCH DECODER-SIDE DEPTH ESTIMATION IN MPEG IMMERSIVE VIDEO

Marta Milovanović[⋆§] *Félix Henry*[§] *Marco Cagnazzo*[⋆] *Joël Jung*[‡]

[§] Orange Labs, France
[⋆] LTCI, Télécom Paris, Institut polytechnique de Paris, France
[‡] Tencent Media Lab, CA, USA

ABSTRACT

This paper presents a new approach for achieving bitrate and pixel rate reduction in the MPEG immersive video coding setting. We demonstrate that it is possible to avoid the transmission of some depth information in the Test Model for Immersive Video (TMIV) by estimating it at the receiver's side. Although the transmitted information in TMIV is considered as non-redundant, we show that it is possible to improve this algorithm. This method provides 3.4%, 9.0%, and 12.1% average BD-rate gain for natural content on high, medium, and low bitrate, respectively, with up to respectively 12.3%, 16.0%, and 18.4% peak reductions. Moreover, it preserves the perceptual quality as measured with MS-SSIM and VMAF metrics. Additionally, it decreases the pixel rate by 8.3% for each test sequence.

***Index Terms*—** immersive video coding, MPEG-I, decoder-side depth estimation, TMIV

1. INTRODUCTION

Immersive video allows a viewer to freely navigate and change the viewpoint inside a 3D scene [1], [2]. A common format for immersive video representation is Multiview Video plus Depth (MVD), where the 3D scene is captured with multiple cameras [3]. In this format, the geometry information of the scene is given by the depth maps of each camera. Furthermore, immersive video needs significantly more data to ensure that the viewer has an adequate depth perception of the scene, compared to classical 2D video. Since capturing the views using a large number of cameras is impractical, one could resort to image synthesis in order to artificially increase the number of available points of view. The most popular image synthesis techniques are based on Depth Image Based Rendering (DIBR) [4]. DIBR methods synthesize arbitrary intermediate views at the receiver side, using the source views and their depth maps. Due to the increasing demand for immersive video consumption, efficient compression and transmission of immersive media became an important task for standardization organizations. The MPEG-I project ISO/IEC 23090, called "Coded Representation of Immersive Media" intends to provide a set of standards to enable immersive media experience. One of the standards that are under development is ISO/IEC 23090 Part 12, called MPEG Immersive Video (MIV) [5]. MIV is a profile of the Visual Volumetric Video-based Coding (V3C), and is based on V3C standard. This standard utilizes 2D video codecs for compression of the source texture and depth information, which are pre-processed before compression. However, in comparison to traditional video coding, immersive video coding is considerably more demanding in terms of complexity. In addition to the trade-off between bitrate and quality, immersive video coding is bounded by the number of decoders that are allowed to run in parallel at the client's side. Moreover, it is constrained by the pixel rate, *i.e.* the number of pixels needed to be decoded per second to present the target view to the user, which became an important factor for mobile use cases.

One of the investigated approaches in MPEG-I is the so-called MV-HEVC + VVS anchor. MV-HEVC is the abbreviation for Multiview extension of High Efficiency Video Coding standard [6], [7]. VVS stands for Versatile View Synthesizer, a synthesis software which was adopted by the MPEG-I Visual group as a reference software for exploration experiments [8]. This approach has the following idea: send many views, consisting of texture and depth map components, and encode them with inter-view prediction, to remove redundancies. The problem with such an approach lies in the pixel rate constraints and the maximum number of simultaneous decoders.

MIV provides another framework that abides the aforementioned constraints. It is based on HEVC, and implemented by the TMIV, Test Model for Immersive Video [9]. TMIV is currently being updated and improved at each MPEG meeting cycle. This software involves many challenging tasks: selecting the most important views among source views (view labeling), removing the redundancies between the views that were not selected (pruning), constructing the atlases that will be sent (packing), and rendering the target viewport (as a part of the decoder). Atlas is defined as a set of 2D rectangles from different views, *i.e.* patches, projected into a rectangular frame [5]. The pruning process decides which pixels of the views to send, *e.g.* the parts that are missing because they are occluded in basic and other additional views. This introduced the concept of patch, which is a rectangular region that is recognized as important for the target view reconstruction. However, the rectangular region is multiplied with the pruning binary mask, which refines the patch shape, and save only the necessary pixels. The process of pruning and packing significantly reduces the bitrate and pixel rate. Nevertheless, the pruning is highly dependent on the depth map quality. The method proposed in this paper is based on the idea that the TMIV pruning method sometimes results in a sub-optimal patches selection.

A different framework for immersive video can be considered. This approach is based on reducing the depth map transmission from the encoder side, and on moving a part of the depth estimation process to the decoder side. Recently, it has been shown that decoder-side depth estimation (DSDE) can efficiently recover the depth maps at the decoder side while reducing the pixel rate by 50% and saving 37.3% in terms of Bjøntegaard delta (BD) rate metric [10], when applied to the aforementioned MV-HEVC + VVS framework [11], [2]. Hence, the DSDE approach was studied in the case of full views, where the depth estimation process took advantage of many views that are available at the decoder. However, if one wants to apply the DSDE ideas in the case of the TMIV framework, some difficulties arise, because the number of available views at the decoder side in TMIV is drastically reduced. The goal of this paper is to explore the

idea of reducing the transmission of the depth data in the context of TMIV. More precisely, we investigate the assumption that it is possible to avoid the transmission of some patch depths that originate from the additional views while saving the bitrate, pixel rate, and preserving the quality. The patches from additional views that are sent as small rectangular areas are of great importance for the rendering process in TMIV, as they are a more efficient way to transmit the non-redundant areas, instead of sending the full views. We want to preserve the patch textures and use them together with the basic view textures to estimate some depth patches at the receiver side, while not changing the pruning process. Thus, we are inspecting the isolated impact of the patch level depth recovery at the decoder side. Following the common test conditions (CTC), defined in January 2020 [12], with small modifications, on 8 perspective test sequences we observe an average BD-Rate Y-PSNR gain of 1.8% for medium bitrate (with gains up to 16.0%), and average of 7.7% for low bitrate (with gains up to 18.4%). Moreover, we achieve 9.3% VMAF [13] BD-Rate reduction for medium bitrate, and 13.4% for low bitrate, while having 6.7% perceptual MS-SSIM [14] gain for medium bitrate, with 11.6% gain for low bitrate. In addition, in all the cases we achieve 8.3% pixel rate reduction.

The rest of this paper is organized as follows. Section 2 gives an overview of the TMIV framework and describes the proposed method. Section 3 presents the objective results and comparison against the anchor. The discussion on the obtained results is given in Section 4. Finally, Section 5 draws the conclusion on this paper.

2. PROPOSED METHOD

2.1. TMIV framework description

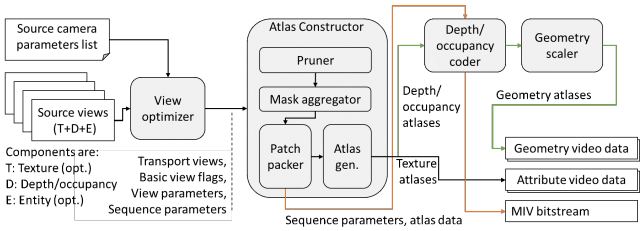


Fig. 1. Encoding flow in TMIV version 4 [9].

In this work we use TMIV 4, established during the 129th MPEG meeting in January 2020 [9]. TMIV is a software that operates with the MVD format, where the content can be natural or computer-generated, and created with omnidirectional or perspective cameras. It receives source views with texture and depth components, as well as the source camera parameters list. The grouping process in TMIV creates subsets of input views. The TMIV encoder can operate on a group basis, thus independently per group. Subsequently, it proceeds to the view labeling process, where the goal is to identify one or multiple views that will be fully transmitted: they are referred to as basic views. The next process is atlas construction, which consists of the pruning, mask aggregation, patch packing, and atlas generation. In this process, multiple views are processed, their redundancies are pruned, and the resulting patches are packed into the atlases. This means that each atlas can contain the patches from multiple views. The key elements of the TMIV encoding flow are shown in Fig. 1.

Pruning is very important since it directly decides which pixels are essential, by establishing the hierarchy of views. Starting from the basic views, it tries to synthesize the views from the top to the bottom of the hierarchy. If the ratio between synthesized and source depth in a pixel is less than a defined threshold, the pixel may be pruned. The pruning mask is computed during each intra period, and reset afterwards. It is used in the patch packing process, to pinpoint the patches. Depth occupancy coder adds occupancy coding information for preserved pixels into the depth atlases. In addition, geometry (depth) scaler implements the down-scaling of the depth map atlases. The output of the TMIV encoding process consists of: texture atlases (attribute video data), depth map atlases (geometry video data), and the MIV bitstream, as shown in Fig. 2. The TMIV decoder incorporates the HEVC video decoders, the MIV normative decoder and metadata parser, and a block to patch map decoder. Besides the normative part, the TMIV decoder consists of the geometry (depth) upscaler, a culler, and a renderer. The TMIV renderer is a DIBR method, called view weighting synthesizer [9]. It takes both the recovered full views and pruned views as input.

2.2. Omitting the depth patch transmission in TMIV

In Fig. 2, we depict the proposed approach by comparison to the anchor: the scheme describes the anchor when the switch is in position 1 and our method when it is in position 0. Note that we consider only the data available at the decoder (receiver), which consists of atlases with basic views, atlases with patches from additional views (patch-atlases), and metadata. Consequently, the decoded atlases have compression artifacts. Initially, the process of view "unpacking" is done, where each texture patch from chosen patch-atlas is projected to the corresponding view. This way, we recover the pruned textures, which we subsequently use in the depth estimation process together with the basic view textures. Following the process of "unpacking", the recovered pruned textures and basic view textures are given to the Immersive Video Depth Estimation (IVDE) software [15]. IVDE is a reference software for exploration experiments adopted by MPEG-I. We chose this software because it is agnostic to the number and positioning of the cameras and it ensures high quality of estimated depth maps, with inter-view and temporal consistencies [16]. IVDE performs the depth estimation on segments, which results in correspondence between the object edges in depth maps and the object edges in input textures, consequently enhancing the synthesis quality. No special adaptation was applied to the IVDE software to warn the estimator about the fact that the input textures contain a significant amount of non-valid pixels (those for which no patch has been transmitted).

Let us denote the source textures as $\mathbf{T}$ and source depth maps as $\mathbf{D}$. All texture atlases are transmitted, while for depth atlases we have the following: anchor sends all depth atlases, while our method sends all basic depth atlases, but avoids sending one depth patch-atlas. The two remaining depth patch-atlases are sent. Fig. 2 is simplified to show only one texture and depth patch-atlas, although there are multiple ones in our setup. At the decoder side, the renderer performs the "unpacking": a projection from the atlases to the corresponding positions in the views. Recovered textures are denoted as $\mathbf{T}^*$ and recovered basic view depths are denoted as $\mathbf{D}^*_\mathbf{B}$. In the anchor case, all depth patch-atlases are sent, and recovered patch depths, denoted as $\mathbf{D}^*_\mathbf{P}$, are used in the rendering process. In our case, the decoding process is done in two stages. First, it recovers the textures $\mathbf{T}^*$ and the depth maps $\mathbf{D}^*_\mathbf{B}$. Then, the textures $\mathbf{T}^*$ are given to the IVDE and used to produce patch depths $\mathbf{D}^{'}_\mathbf{P}$. After obtaining the $\mathbf{D}^{'}_\mathbf{P}$, the rendering of the target viewport is continued.

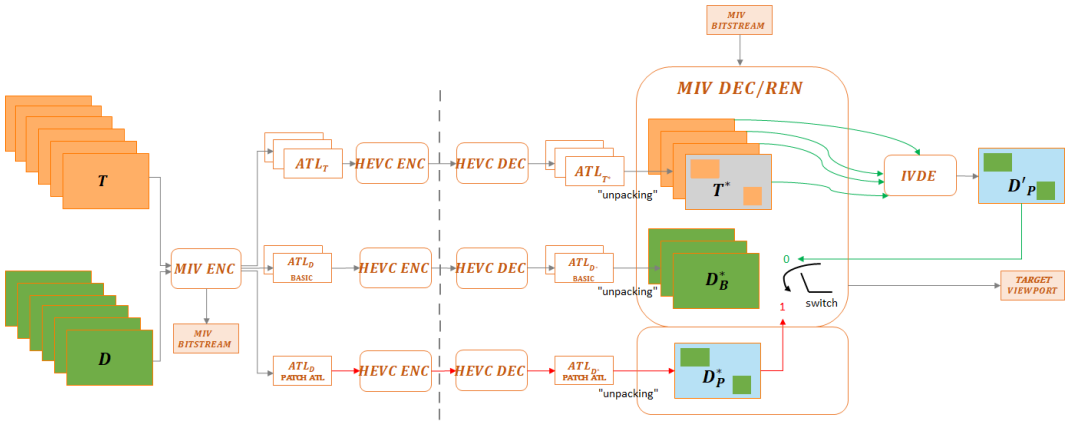


Fig. 2. Process diagram for the anchor ($switch = 1$) and our method ($switch = 0$) in the TMIV framework.

3. EXPERIMENTAL RESULTS

3.1. Test conditions

Our experimental setting complies with the MPEG methodology and common test conditions (CTC) defined in January 2020 [12], except for the following: the atlases are constructed with the same size as the source videos, and the depth atlases are not down-sampled. These changes are made to facilitate the experimental process and the comparison with the anchor. Each sequence was encoded in the setup of three groups, where each group has one atlas containing a basic view and one patch-atlas. Furthermore, in addition to the five given quantization parameter pairs (QP_T, QP_D) for video compression with HEVC Main 10 profile, another quantization parameter pair is added to show the performance at low bit-rate. The set of (QP_T, QP_D) pairs is: $\{(22,4),(27,7),(32,11),(37,15),(42,20),(47,25)\}$. The first four pairs are used for high bitrate, the second to fifth for medium bitrate, and the last four for low bitrate. The CTC defines test sequences used for the evaluation of proposed algorithms. Our approach was tested on eight perspective sequences with HD and 2K resolutions, two of which are computer generated (CG), and six of which are natural content (NC). All of the used sequences are perspective type content, while omnidirectional content was not tested in the scope of this work. Omnidirectional CTC sequences have much higher resolutions which leads to having only one basic view at the decoder side, reducing the depth estimation to the case with stereo input. On top of that, the baselines between the views are larger than in the regular depth estimation scenario. Since IVDE depth estimation software was used for the local estimation of patch depths without modifications, which is already a delicate process, it was decided to limit the test set to perpective content. We evaluated the bitrate and synthesized view quality performance provided by the proposed method compared to the anchor, using the Bjøntegaard delta rate metrics, in terms of Y-PSNR, VMAF, and MS-SSIM.

3.2. Results

For each sequence and each QP_T value, one patch-atlas with multiple patches was "unpacked" and saved as a set of the recovered pruned textures. This method was tested for each depth patch-atlas individually, and the one with the best performance was chosen. The corresponding depth atlas was not transmitted and patch depths were replaced with the depths obtained with IVDE from decoded textures

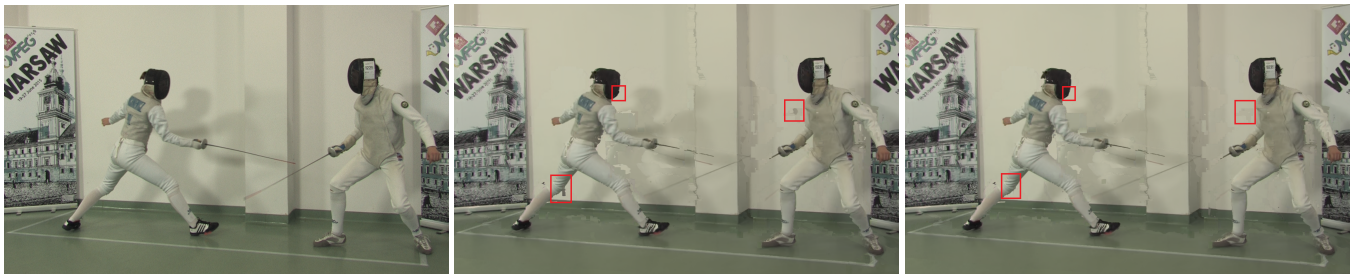
Sequence	CTC - High bitrate	CTC - Medium bitrate	Low bitrate
Shaman (CG)	26.34	8.99	0.73
Kitchen (CG)	66.95	30.33	10.72
Painter (NC)	2.75	-7.81	-12.86
Frog (NC)	3.57	-3.49	-8.01
Fencing (NC)	-12.33	-16.02	-18.35
Carpark (NC)	0.26	-8.33	-12.63
Street (NC)	-6.10	-8.67	-10.65
Hall (NC)	-8.53	-9.48	-10.17
Average (all)	**9.11**	**-1.81**	**-7.65**
Average (NC)	**-3.40**	**-8.97**	**-12.11**

Table 1. BD-rate results per test sequence, in terms of Y-PSNR of synthesized texture [%]. Negative values indicate gains.

Sequence	VMAF			MS-SSIM		
	High	Med	Low	High	Med	Low
Shaman (CG)	3.44	-6.69	-11.53	20.60	1.54	-6.33
Kitchen (CG)	46.96	12.77	-0.28	20.14	5.08	-3.38
Painter (NC)	-13.04	-18.59	-21.31	-4.21	-13.96	-18.22
Frog (NC)	-3.57	-8.57	-11.13	6.19	-5.49	-10.07
Fencing (NC)	-12.91	-16.87	-19.60	-3.74	-13.29	-17.03
Carpark (NC)	-5.53	-13.14	-16.52	-1.70	-12.27	-16.15
Street (NC)	-7.00	-9.52	-11.32	-6.54	-9.31	-11.29
Hall (NC)	-10.84	-13.35	-15.09	3.93	-5.73	-10.22
Average (all)	**-0.31**	**-9.25**	**-13.35**	**4.33**	**-6.68**	**-11.59**
Average (NC)	**-8.82**	**-13.34**	**-15.83**	**-1.01**	**-10.01**	**-13.83**

Table 2. BD-rate results per test sequence, in terms of VMAF and MS-SSIM metrics [%]. Negative values indicate gains.

per each QP_T, at the decoder side. The obtained results are shown in Table 1 and Table 2. Negative values indicate BD-rate gains, while positive values indicate losses. The data show BD-rate losses for CG sequences on high and medium bitrate range that diminish with the increase of QP parameters. Moreover, the trend of bigger BD-rate gains as QPs increase is constant for all the sequences. In addition, this method yields an 8.3% pixel rate reduction per sequence. Fig. 3 demonstrates how some artifacts can be avoided by not sending some of the patch depths (marked with red rectangles). In this case,



(a) Source texture. (b) Synthesized view: anchor. (c) Synthesized view: proposed method.

Fig. 3. One of the views in the Fencing sequence, compared to the anchor synthesis result, and synthesis result of the proposed method.

the synthesis result of the proposed method subjectively seems better. Fig. 4 compares the rate-distortion (RD) curves of the anchor and proposed method for the Fencing test sequence. Our method performs better on the whole bitrate range.

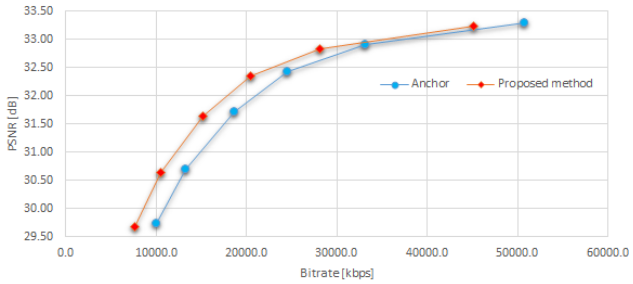


Fig. 4. RD curves for the Fencing sequence.

The fraction of the depth in the bitrate (it consists of texture and depth component) varies depending on the nature of the depth map. CG sequences, Kitchen and Shaman, as well as NC sequence Frog, have a depth fraction from 30% on high bitrates, up to 70% on low bitrates. Other NC sequences take from 50% on high bitrates, up to 85% on low bitrates.

4. DISCUSSION

In this paper, we have shown that it is possible to avoid the transmission of some patch depths, and estimate them at the receiver side, while preserving the quality of synthesized views. The TMIV system is very complex with many interconnected blocks. Despite that, we introduce a new paradigm by not sending the significant amount of patch depths, and we demonstrate BD-rate gains for natural content. We observe that the results are very consistent, for the majority of the natural content. The bitrate reduction is significant, as given in an example in Fig. 4. For each sequence, the BD-rate results gradually improve towards low bitrate, which indicates that our method would have a satisfactory streaming performance when network bandwidth is limited. Furthermore, this method provides a pixel rate reduction of 8.3% per sequence, which is very important for some use cases, *e.g.* mobile devices.

The losses on the computer generated content can be explained if we take into account the nature of considered depth maps. Since CG depth maps are ground-truth depth maps, generated with mathematical models of the captured 3D scene, the hypothesis on which pruning is based is perfectly met. Nevertheless, the depth maps for natural content, obtained by a sensor or some depth estimation algorithm, are not perfect, which creates room for the improvement of the pruning method. Some source depth maps in our test set are obtained using IVDE, while others are computed with similar tools, all of which are exposed to additional post-processing methods. In the case of CG content from our test set, the fraction that belongs to depths in the bitstream is significantly lower compared to depth fraction of the NC sequences. As a consequence, the amount of overall saved bitrate for CG content with our method is reduced. Moreover, looking at the patch-atlases of the tested CG content, we can notice that the majority of the patches have homogeneous textures, which is very unfavorable for depth estimation algorithms. In addition, our CG content has significantly more source views (25) than the natural content $(9-16)$. This results in the sparser sampling of texture patches which are preserved in TMIV during pruning. Reduced texture information then leads to deterioration of the quality of estimated depth maps. NC sequences that had BD-rate Y-PSNR losses on high bitrate are Painter, Frog and Carpark. Despite that, they demonstrated good performance in terms of preserving the perceptual quality, as measured by VMAF in Table 2.

Depth estimation is a delicate process that aims to find the correspondence in two or more views, while pruning is built to eliminate the areas which have some correspondence with the other views. Therefore, when we disable the patch depth transmission, we should change the pruning strategy accordingly to ensure a reliable depth estimation at the decoder side. However, in this article, we chose to observe only the isolated impact of the patch depth recovery without modifying the pruning. Aside from pruning strategy, we are facing more challenges: depth estimation from compressed textures and local depth estimation on very small patch areas. The obtained results are important because they show that proposed method improves the TMIV coding system despite the above-mentioned challenges.

5. CONCLUSION

This paper presents a novel approach for tackling the bitrate and pixel rate constraints in immersive video coding. It is a proof-of-concept of the idea which relies on omitting the transmission of some depth patches in TMIV. Consequently, the patch depth estimation is done at the decoder side, using decoded basic views and patches. We present proposed setting and show BD-rate savings on natural content, for high, medium and low bitrate, in the terms of Y-PSNR, VMAF, and MS-SSIM metrics. However, our study shows that enabling the decoder-side patch depth estimation is a challenging task, especially for computer generated content. To address this problem, a different pruning strategy should be considered in the future.

6. REFERENCES

[1] Frederic Dufaux, Béatrice Pesquet-Popescu, and Marco Cagnazzo, *Emerging technologies for 3D video: creation, coding, transmission and rendering*, John Wiley & Sons, 2013.

[2] Masayuki Tanimoto, "Overview of FTV (free-viewpoint television)," in *2009 IEEE International Conference on Multimedia and Expo*, New York, NY, USA, June 2009, pp. 1552–1553, IEEE.

[3] Philipp Merkle, Aljoscha Smolic, Karsten Muller, and Thomas Wiegand, "Multi-View Video Plus Depth Representation and Coding," in *2007 IEEE International Conference on Image Processing*, San Antonio, TX, USA, Sept. 2007, pp. I – 201–I – 204, IEEE, ISSN: 1522-4880.

[4] Christoph Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," San Jose, CA, May 2004, pp. 93–104.

[5] "Working Draft 4 of Immersive Video," ISO/IEC JTC 1/SC 29/WG 11 N19001, Feb. 2020.

[6] Gary J. Sullivan, Jill M. Boyce, Ying Chen, Jens-Rainer Ohm, C. Andrew Segall, and Anthony Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[7] Gerhard Tech, Ying Chen, Karsten Muller, Jens-Rainer Ohm, Anthony Vetro, and Ye-Kui Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[8] Joël Jung and Patrick Boissonade, "VVS: Versatile View Synthesizer for 6-DoF Immersive Video," working paper or preprint, Apr. 2020.

[9] Basel Salahieh, Bart Kroon, Joel Jung, and Marek Domański, "Test Model 4 for Immersive Video," ISO/IEC JTC 1/SC 29/WG 11 N19002, Feb. 2020.

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

[11] Patrick Garus, Joel Jung, Thomas Maugey, and Christine Guillemot, "Bypassing Depth Maps Transmission For Immersive Video Coding," in *2019 Picture Coding Symposium (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5, IEEE.

[12] Joel Jung, Bart Kroon, and Jill Boyce, "Common Test Conditions for Immersive Video," ISO/IEC JTC 1/SC 29/WG 11 N18997, Feb. 2020.

[13] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, vol. 6, pp. 2, 2016.

[14] Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Pacific Grove, CA, USA, 2003, pp. 1398–1402, IEEE.

[15] Dawid Mieloch, Adrian Dziembowski, Jakub Stankowski, Olgierd Stankiewicz, Marek Domański, Gwangsoon Lee, and Yun Young Jeong, "Immersive video depth estimation," ISO/IEC JTC 1/SC 29/WG 11 m53407, Apr. 2020.

[16] Dawid Mieloch, Olgierd Stankiewicz, and Marek Domanski, "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," *IEEE Access*, vol. 8, pp. 5760–5776, 2020.