HAL open science

Depth Patch Selection for Decoder-Side Depth Estimation in MPEG Immersive Video

Marta Milovanovic, Felix Henry, Marco Cagnazzo

To cite this version:

Marta Milovanovic, Felix Henry, Marco Cagnazzo. Depth Patch Selection for Decoder-Side Depth Estimation in MPEG Immersive Video. 2022 Picture Coding Symposium (PCS), IEEE, Dec 2022, San Jose, United States. pp.343-347, 10.1109/PCS56426.2022.10018042 . hal-04533753

HAL Id: hal-04533753

https://hal.science/hal-04533753v1

Submitted on 5 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Depth Patch Selection for Decoder-Side Depth Estimation in MPEG Immersive Video

Marta Milovanović
LTCI, Télécom Paris, IP Paris
Orange Labs
Guyancourt, France
marta.milovanovic@telecom-paris.fr

Félix Henry
Orange Labs
Cesson-Sévigné, France
felix.henry@orange.com

Marco Cagnazzo
LTCI, Télécom Paris, IP Paris
DEI, University of Padua
Padua, Italy
marco.cagnazzo@telecom-paris.fr

***Abstract*—The MPEG immersive video (MIV) standard has been developed to efficiently compress volumetric video content and enable an immersive user experience. MIV deals with an enormous amount of data that comes in the form of multi-view plus depth videos, which is efficiently reduced in the process of pruning, by tackling the redundancies among the views. This paper presents a novel approach for improving the existing immersive video coding scheme. The proposed approach reduces the amount of transmitted depth data, leveraging the fact that the depth information is partially contained in texture videos. The study proposes a method that ensures a reliable recovery of depths at the decoder-side. This method provides BD-rate improvements on both high and low bitrate ranges, with up to 22.57% Y-PSNR, 25.76% VMAF, 24.07% MS-SSIM, and 22.94% IV-PSNR metric gain, given a low bitrate setting.**

***Index Terms*—immersive video coding, MPEG, MIV, depth map, video processing**

I. Introduction

Recently, the popularity of virtual reality, augmented reality, extended reality, and metaverse commodities surged. These emerging use cases induced new challenges in the area of data compression for enabling data transmission over current networks. The Moving Picture Experts Group (MPEG) is addressing these challenges in the scope of the MPEG-I project [1]. One of the standards from this project is ISO/IEC-23090 part 12 called MPEG Immersive Video (MIV) standard [2], [3], which is focused on the transmission of multi-view video plus depth (MVD) data [4]. MVD is one of the most popular formats for facilitating free navigation of a user in a limited volume, with six degrees of freedom (6DoF) [5], [6]. In this context, the real or virtual 3D scene (natural or computer-generated content) is captured by multiple real or virtual cameras, with an arbitrary arrangement, and each viewpoint has its own texture video and corresponding depth map. The target viewport is rendered from available information at the decoder-side utilizing depth image-based rendering (DIBR) techniques [7], [8]. Such dense sampling of the scene and high-resolutions, needed to ensure a satisfactory immersive experience, comprise a vast amount of data that, due to its size, cannot be transmitted over the networks. Therefore, an efficient algorithm is essential to eliminate the redundancies among the videos of the captured scene.

Previously, the standards aiming to achieve multi-view video compression, such as MV-HEVC and 3D-HEVC [9] were not widely adopted due to their limitations and impracticalities (large prediction structures, ability to deal only with coplanar content with a narrow baseline, necessity to externally define the reference views). As opposed to these standards, the concept of MIV and its corresponding Test Model for MPEG Immersive Video (TMIV) [10] is not to deploy inter-view prediction and depth coding tools but rather to extract from MVD-captured data, a minimal subset of texture and depth information allowing to reconstruct the 3D scene. More precisely, the idea is to find and prune the redundancies among the videos, *i.e.*, to identify points that are available in numerous views and encode them only once: identify the patches, pack them into atlases, and compress the resulting partial videos with any legacy 2D video codec, such as HEVC or VVC. Therefore, the benefits of MIV are that it is codec agnostic, can handle any camera setup, reduces the bitrate with respect to the full MVD data, and also reduces the pixel rate, *i.e.* the number of pixels needed to be decoded per second in order to render the target virtual view. On the other hand, TMIV depends on two important non-normative stages: depth estimation (in the case of natural content) and view synthesis.

Since bitrate and pixel rate constraints are crucial for an immersive video end-to-end transmission system, there have been a lot of improvements and evolution of TMIV in these aspects. The pruning process performs pixel-matching among all possible pairs of views. It prunes a pixel in one view if it finds a similar pixel in another (which has the same position in 3D space), while it preserves pixels that are not visible in other views [11], [12]. Depth map quality has a significant impact on rendering. However, depth maps have a lot of spatial redundancy and it is burdensome to compress them with conventional 2D video codecs. Thus, the spatial downsampling of depth maps is done in the TMIV processing chain. Furthermore, a new approach emerged, the so-called decoder-side depth estimation (DSDE) [13], [14], which improved the immersive video system by entirely avoiding the transmission of depth maps and moving the depth estimation process to the decoder-side. This concept was adopted in the MIV standard as Geometry Absent (GA) profile [15]. The

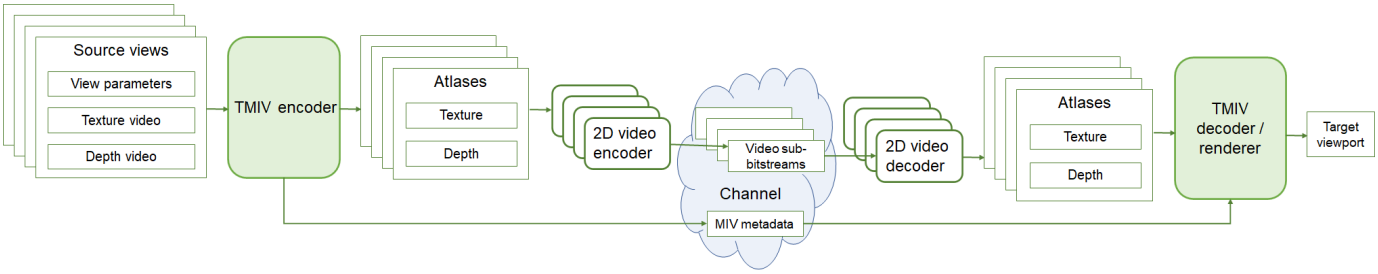

Fig. 1: MPEG immersive video transmission scheme.

DSDE approach was studied for the case of fully transmitted views (without pruning), where the depth estimation process is straightforward. Our previous work [16] moved the aforementioned concept further by showing that it is possible to avoid the transmission of pruned depths and recover their depth maps at the decoder-side while preserving the synthesis quality. This approach is "blind" in the sense that it chooses not to send one from a few available patch depth atlases, without any quality-based criteria, and it yields significant gains over the anchor, especially on low bitrates.

This study proposes an improvement compared to the "blind" patch-DSDE approach. We hypothesize that it is possible to reliably discriminate between the depth patches that have to be sent and those that can be estimated at the decoder, based on the quality of estimated depth patches at the encoder-side. We omit their transmission and subsequently recover them at the decoder-side using a depth estimator, such as Immersive Video Depth Estimation (IVDE) [17]. IVDE is a reference software for depth estimation, convenient in the use case of immersive video because it is agnostic to the camera arrangement and number of cameras, and produces high-quality depth maps. The results show significant gains in comparison to the TMIV-coded anchor: on low bitrate, an average BD-rate gain of 4.63% for Y-PSNR, 6.21% for VMAF, 5.70% for MS-SSIM, and 4.98% for IV-PSNR, and on high bitrate: 2.03% for Y-PSNR, 4.05% for VMAF, 3.15% for MS-SSIM, and 2.28% for IV-PSNR metric. The remainder of the paper is structured as follows. Section 2 gives a brief overview of the TMIV processing algorithms. Section 3 demonstrates our proposed improvement. Section 4 presents experimental conditions, results, and their analysis, whereas Section 5 concludes the paper.

II. Background

TMIV is a reference software of MIV, whose immersive video transmission scheme is illustrated in Fig. 1. It carries out preprocessing of the source views, detecting essential samples and packing them as patches into texture and depth atlas videos. Moreover, it also produces the MIV bitstream, containing the metadata required to decode all information about the patches that are necessary for the rendering of the desired viewport.

The TMIV encoder can encode the source views in one or multiple groups (subsets), enabling partial decoding at the decoder. Source views are labeled either as "basic" or "additional". This results in basic views being packed into atlases and transmitted in their integrity, whereas the samples from additional views are processed further in the redundancy removal process, so-called pruning. After determining which patches are essential for a good rendering of a virtual view at the decoder side, patches are packed into atlases. The pruning binary mask contains the information if the pixel is preserved or pruned, and it is aggregated over an intra-period, to ensure the temporal consistency of a rendered video. Optionally, depth atlases can be downsampled and quantized.

The pruning process is a fundamental tool in TMIV for removing redundant samples among the source views. It determines which pixels in additional views are already present either in the basic views or other additional views and therefore not needed for transmission; on the contrary, pixels that cannot be recovered from other views, are preserved. Firstly, a pruning graph is created, establishing the pruning hierarchy, having basic views as roots. All additional views are successively added to the graph as child nodes of the already present nodes. Then, a pixel is pruned if it fulfills the following criteria:

- The pixel is synthesized from the views higher up in the hierarchy.
- The difference between synthesized and source geometry is in a defined range.
- The difference between synthesized and source luma co-located block is in a defined range.

Additionally, a second-pass pruning step based on global color matching is introduced. Its goal is to restore some of the pruned pixels, which were initially pruned due to depth errors or illumination changes among the source textures. Finally, temporal consistency is achieved with the pruning mask aggregation process.

The TMIV decoding process starts with a user's request for the desired viewport, by giving its coordinates. The TMIV decoder invokes the 2D video sub-bitstream decoder, MIV metadata parser, and block to patch map decoder. This normative process is followed by a non-normative rendering of the target viewport.

III. Proposed Method

This section gives an overview of our proposal. It is a hybrid patch-DSDE approach, where a reliable, quality-based depth patch selection is done on a per-patch basis at the encoder

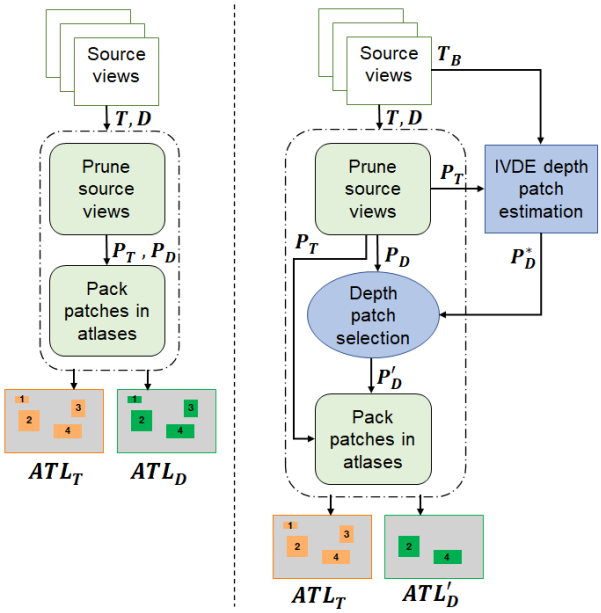

Fig. 2: Left: TMIV anchor encoding of additional views. Right: Proposed method for depth patch selection in additional views.

side. The decision (send / do not send) is transmitted to the decoder as a flag in the bitstream. The idea of the proposed scheme is to improve the reference method by ensuring that no redundant information is transported. Knowing that depth information is somewhat present in textures, we hypothesize that some depths can be omitted and afterwards recovered at the decoder-side.

A. Depth patch selection

Our method for depth patch selection is shown in Fig. 2, as compared to anchor TMIV encoding. This scheme is simplified to illustrate the pruning and packing process in the case of an atlas with patches originating from additional views. First, source views are encoded with TMIV in the "MIV main" anchor mode. Afterwards, the uncompressed basic texture views T_B and pruned "anchor" textures P_T and "anchor" depths P_D views are stored. Then, the depths P_D^* are estimated from the uncompressed textures T_B and P_T. All depth views are estimated, except for basic view depths, which are to be transmitted without modifications. The estimated depth patches, obtained from the estimated depths P_D^*, are then compared with the "original" depth patches, obtained from depths P_D, by computing the PSNR: if an estimated depth patch is sufficiently similar to the original one (*i.e.,* the obtained PSNR is larger than a given threshold T_H), we argue that we can avoid sending that patch depth. Only the patch depths that cannot be estimated with sufficient quality need to be sent ($P_D^{'}$ in Fig. 2). A flag is stored per patch to convey this information to the modified decoder. The threshold T_H is sequence-dependent since it depends on the quality of the estimated depth patch, which varies across different patches and sequences.

TABLE I: Test sequences and their parameters.

Sequence	Type	Resolution	#Views
Painter	NC	2048×1088	16
Frog	NC	1920×1080	13
Carpark	NC	1920×1088	9
Fan	CG	1920×1080	15
Kitchen	CG	1920×1080	25
Fencing	NC	1920×1080	10
Hall	NC	1920×1088	9
Street	NC	1920×1088	9
Mirror	CG	1920×1080	15

This study is an improvement of the "blind" patch-DSDE approach in [16]: instead of omitting some amount of patch depths without any depth or rendered view guarantee of quality, we ensure that the omitted patch depths are adequate to be estimated at the decoder side. Nevertheless, this approach comes with some inconveniences. First, the transmitted flag is an additional cost to the bitstream, although it is negligible. Besides that, the depth patch selection method is heuristic. Ideally, one should render all possible target viewports using the estimated depths and compare them with the viewports rendered using the original depths. Since this approach is unfeasible because of its complexity, we propose a suitable proxy (depth quality comparison) as a compromise.

B. Patch decoder-side depth estimation

The decoding process of the proposed method is similar to the one described in [16]. First, the atlases are decoded and all the transported views are recovered (basic and additional). The TMIV decoder is modified to read the flag which signals if a depth patch is transmitted or not. If a certain depth patch was not transmitted, the process for patch decoder-side depth estimation is invoked at the decoder-side, where the depth patch is estimated with IVDE software from all available decompressed textures: basic views and the corresponding texture patch. The estimated depth patches are written to the corresponding positions of the recovered views. Thus, these estimated depths are used alongside the transported source depths during the rendering process.

IV. Experimental Results and Discussion

A. Test conditions

To evaluate our approach and to ensure a fair comparison, we follow the common test conditions (CTC) as defined by MPEG-I [18]. We compare the proposal with the anchor on nine perspective sequences, both natural and computer-generated. In both cases, the depth maps have been generated by IVDE. A summary of test sequences' characteristics is given in Table I. In our study, we used TMIV7 in "MIV main" mode, which processes the source views (pruning, packing) to generate the atlases. Atlas videos are compressed and decompressed with HEVC (HM16.16), using five different rate points and corresponding quantization parameters QPs

as defined by the CTC. The pixel rate constraints are the following:

- The combined luma sample rate in all decoders does not surpass 1 069 547 520 samples per second (according to HEVC Main 10 profile level 5.2).
- Maximum luma picture size of each coded video does not surpass 8 912 896 pixels (*e.g.* 4096 × 2048).
- The maximum number of decoder instantiations is four.

TMIV automatically computes the atlas frame sizes based on given pixel rate constraints. For decoder-side patch depth estimation in our approach, we used IVDE4 software. To enable evaluation of the results of novel view synthesis using objective metrics, TMIV renders output views in corresponding positions of source views. We evaluated the rendered view quality performance provided by our approach compared to the MIV anchor, with Bjøntegaard delta (BD) rate metrics [19], [20] computed over the four largest QPs (low bitrate) and over the four smallest QPs (high bitrate). BD-rate is calculated in terms of Y-PSNR, VMAF [21], MS-SSIM [22], and IV-PSNR [23], where IV-PSNR is a metric adapted for specific characteristics of immersive video. The presented synthesis results are averaged over all views, for each QP. The evaluation is done on 17 frames.

B. Results

The obtained results are shown in Table IIa, for low bitrate range, and in Table IIb, for high bitrate range. Negative values indicate BD-rate gains, whereas positive values indicate losses of the proposed method compared to the anchor. The data show average gains on all computed metrics and for both bitrate ranges, with low bitrate peak gains of 22.57% for Y-PSNR, 25.76% for VMAF, 24.07% for MS-SSIM, and 22.94% for IV-PSNR. The peak gains on high bitrate range yield 15.20% for Y-PSNR, 18.70% for VMAF, 13.16% for MS-SSIM, and 12.87% for IV-PSNR. Our method performs particularly well on Frog, Fencing, and Street, while the weakest performance is noticeable on computer-generated content. The fraction of depth maps in the total bitrate is significant, and varies per sequence: 15% to 50% on high and 40% to 80% on low bitrate. On average, our method provides better results for low bitrates, which is coherent with the results obtained in other MIV DSDE studies [15], [16]. Since legacy 2D video codecs are not convenient for depth map compression, using high compression on them causes harmful artefacts and subsequent deterioration of rendered views. Therefore, patch-DSDE approach proves to be beneficial in comparison to the anchor. Furthermore, this method achieves a pixel rate reduction between 0.002% and 5.51% per sequence, which depends on the sequence and the value of its selection threshold T_H (varies from 14 dB to 43 dB). The added flag for depth patch selection gives an average overhead of 0.002% on high and 0.04% on low bitrate.

Fig. 3 illustrates the visual comparison between the anchor and proposal for sequences Painter and Street. As can be seen from these examples, the quality of rendered views in both cases is very similar. Moreover, selected details of the Painter sequence show that colors and object edges are preserved better in the case of the proposed method. Since the VMAF metric seeks to reflect the viewer's perception of the streaming quality, it is no surprise that our method yields significant gains in terms of this metric.

TABLE II: BD-rate synthesis results [%] in terms of Y-PSNR, VMAF, MS-SSIM, and IV-PSNR: (a) low bitrate, (b) high bitrate setting. Negative values indicate gains.

(a) Low bitrate synthesis results.

Sequence	BD-rate Y-PSNR	BD-rate VMAF	BD-rate MS-SSIM	BD-rate IV-PSNR
Painter	-0.20	-0.84	-0.66	-0.26
Frog	-9.57	-13.65	-11.60	-9.69
Carpark	-0.18	-0.47	-0.44	-0.16
Fan	-0.14	-0.12	-0.35	-0.02
Kitchen	-0.07	-0.21	-0.08	-0.06
Fencing	-22.57	-25.76	-24.07	-22.94
Hall	0.00	0.00	0.01	-0.04
Street	-8.92	-14.82	-14.09	-11.67
Mirror	0.00	-0.05	0.00	-0.02
Average	**-4.63**	**-6.21**	**-5.70**	**-4.98**

(b) High bitrate synthesis results.

Sequence	BD-rate Y-PSNR	BD-rate VMAF	BD-rate MS-SSIM	BD-rate IV-PSNR
Painter	0.50	-0.73	-0.47	0.51
Frog	-5.02	-11.62	-8.39	-3.58
Carpark	-0.20	-0.42	-0.35	-0.28
Fan	0.36	-0.02	-0.12	0.20
Kitchen	-0.02	0.04	0.00	-0.06
Fencing	-15.20	-18.70	-13.16	-12.87
Hall	-0.12	0.00	0.01	-0.08
Street	1.27	-4.95	-6.06	-4.48
Mirror	0.17	-0.08	0.20	0.12
Average	**-2.03**	**-4.05**	**-3.15**	**-2.28**

C. Discussion

The main contribution of this paper is in the reliable discrimination between the depth patches needed for transmission and the ones which are redundant, and therefore, possible to recover at the decoder-side. In this study, we have shown that, despite the pruning process, there is some residual redundancy in the set of texture and depth patches. This redundancy can be reduced by omitting the transmission of depth patches that can be accurately estimated at the encoder. More importantly, by imposing an appropriate selection criterion, it is possible to ensure the high quality of estimated depth patches, which consequently results in high-quality rendering.

Threshold: The current version of the method relies on multiple coding passes in order to find the best threshold T_H for patch selection. This results in good rate-distortion performance but also increased complexity. Still, we can reasonably reduce the complexity of these multiple passes. An extensive analysis of the threshold per sequence is needed because we do not have an a priori knowledge of estimated depth quality. However, after the first analysis, it is possible to update the threshold solely on a periodical basis, *e.g.,* when characteristics of a sequence significantly change. Thus, this approach is not far from what a practical system could achieve.

(a) Painting sequence. (b) Street sequence.

Fig. 3: Subjective comparison for fragments of rendered views: (a) Painting, (b) Street sequence. Rows: source texture, anchor, and proposed method.

Selection criterion: The best criterion for a DIBR method would be to evaluate directly the quality of rendered views obtained using estimated depth maps, as compared to the rendered views obtained using original depth maps. However, since that would be too complex, we propose to use the quality of the estimated depths as a proxy.

V. Conclusion

This paper proposes a new approach that has been developed to tackle the depth patch redundancies in the MPEG immersive video coding setup. More specifically, a depth patch selection scheme is developed, where the decision for sending a depth patch is made based on the encoder-side depth patch estimation quality, as compared to the source depth patch. Therefore, the non-transmitted depth patches are recovered using compressed textures at the decoder-side. Performance of the proposed method is evaluated under regular common test conditions, on perspective sequences. We show BD-rate savings for high and low bitrates, in terms of Y-PSNR, VMAF, MS-SSIM, and IV-PSNR metrics. However, our method requires multiple coding passes in order to find the best threshold, which increases the encoding complexity. In the future, a different approach for deriving the threshold could be considered.

References

[1] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, "Standardization Status of Immersive Video Coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 5–17, Mar. 2019.

[2] "Text of ISO/IEC FDIS 23090-12 MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N00111, Jul. 2021.

[3] J. M. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, "MPEG Immersive Video Coding Standard," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521–1536, 2021.

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-View Video Plus Depth Representation and Coding," in *2007 IEEE International Conference on Image Processing*. San Antonio, TX, USA: IEEE, Sep. 2007, pp. I – 201–I – 204, iSSN: 1522-4880.

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

[6] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging technologies for 3D video: creation, coding, transmission and rendering*. John Wiley & Sons, 2013.

[7] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," A. J. Woods, J. O. Merritt, S. A. Benton, and M. T. Bolas, Eds., San Jose, CA, May 2004, pp. 93–104.

[8] S. Fachada, D. Bonatto, A. Schenkel, and G. Lafruit, "Depth image based view synthesis with multiple reference views for virtual reality," in *2018 - 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2018, pp. 1–4.

[9] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. 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.

[10] B. Salahieh, B. Kroon, J. Jung, B. Kroon, and A. Dziembowski, "Test Model 7 for MPEG Immersive Video," ISO/IEC JTC 1/SC 29/WG 4 N0005, Oct. 2020.

[11] J.-B. Jeong, S. Lee, D. Jang, and E.-S. Ryu, "Towards 3DoF+ 360 Video Streaming System for Immersive Media," *IEEE Access*, vol. 7, pp. 136 399–136 408, 2019.

[12] H.-C. Shin, J.-Y. Jeong, G. Lee, M. U. Kakli, J. Yun, and J. Seo, "Enhanced pruning algorithm for improving visual quality in MPEG immersive video," *ETRI Journal*, vol. 44, no. 1, pp. 73–84, 2022.

[13] P. Garus, J. Jung, T. Maugey, and C. Guillemot, "Bypassing Depth Maps Transmission For Immersive Video Coding," in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.

[14] P. Garus, F. Henry, J. Jung, T. Maugey, and C. Guillemot, "Immersive video coding: Should geometry information be transmitted as depth maps?" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3250–3264, 2022.

[15] D. Mieloch, P. Garus, M. Milovanović, J. Jung, J. Y. Jeong, S. L. Ravi, and B. Salahieh, "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6360–6374, 2022.

[16] M. Milovanović, F. Henry, M. Cagnazzo, and J. Jung, "Patch Decoder-Side Depth Estimation In Mpeg Immersive Video," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1945–1949.

[17] D. Mieloch, O. Stankiewicz, and M. Domanski, "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," *IEEE Access*, vol. 8, pp. 5760–5776, 2020.

[18] J. Jung and B. Kroon, "Common Test Conditions for MPEG Immersive Video," ISO/IEC JTC 1/SC 29/WG 04 0051, Oct. 2020.

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

[20] S. Pateux and J. Jung, "An excel add-in for computing Bjontegaard metric and its evolution," ITU-T SG16 Q 6, Doc. VCEG-AE07, Jan. 2007.

[21] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, 2016. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

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

[23] A. Dziembowski, D. Mieloch, J. Stankowski, and A. Grzelka, "Iv-psnr – the objective quality metric for immersive video applications," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.