Immersive Video Postprocessing for Efficient Video Coding

Adrian Dziembowski, Dawid Mieloch, *Member, IEEE*, Jun Young Jeong, and Gwangsoon Lee

Abstract—This paper describes two methods for increasing the efficiency of the MPEG Immersive Video (MIV) coding standard. The methods manipulate the MIV-formatted atlas videos by considering the coding principles of the widely used video encoders, enhancing the encoding efficiency for the immersive video content. The first method, patch average color modification removes the constant component of all YCbCr components of each patch within atlases, resulting in the reduction of the number and magnitude of edges within a texture atlas video. The second proposed method changes the dynamic range of the geometry (depth) atlas, adapting to the quality of input depth maps. Both methods proposed by the authors of this paper were included into the Test Model for MPEG Immersive Video (TMIV), which is the reference implementation of the MIV codec. Moreover, the metadata syntax relating to the first proposed method was adopted to the ISO/IEC 23090-12 standard.

Index Terms— immersive video coding, MPEG immersive video, video processing

I. Introduction

In a typical traditional video system, all users are allowed to passively watch an identical video recorded by a specific camera chosen among multiple cameras by the service provider. Whereas in an immersive video system, each user is able to select his or her arbitrary viewpoint within a predefined viewing volume [1]. Thus, to support such a service, the broadcaster needs to transmit a much greater amount of data consisting of all cameras' videos and the corresponding scene's geometry information, preferably within a single bitstream. To efficiently encode these vastly sizeable data, a dedicated encoder is necessary.

One of the most common data representations used for expressing an immersive scene is multiview video which consists of multiple videos and depth maps. In the past decade, several compression methods for multiview video have been proposed, developed, and, very recently, standardized. The most straightforward option is a multiview simulcast coding, where all the videos and depth maps are independently coded with a batch of typical video encoders (e.g., AVC [2], HEVC [3], or VVC [4]). However, this requires high-performance rendering devices that support the simultaneous running of several video decoders, which is generally not possible at consumer-level hardware. More dedicated approaches, such as 3D-HEVC and MV-HEVC [5], take into account inter-view redundancy among different views, but a couple of limitations hinder them to be used in an immersive video system. For instance, 3D-HEVC only works well for linear camera arrangements, and this strictly confines the movement freedom of a user, especially in the rotation aspect. Moreover, this codec supports video only in a perspective format, which is only one of the formats commonly used as immersive video representation. In the case of MV-HEVC, it does not exploit the available depth information for removing spatially redundancy data which is not efficient enough. The problems created a need for a new method for compressing immersive video, which would satisfy these essential requirements.

Recently, several interesting encoding schemes of utilizing the typical video compression methods merged with the pre-processing of encoded video can be seen. Such methods propose changing video that is required to be encoded for a specific use case to be better adjusted to the most widespread video compression methods. This general scheme can be seen, e.g., in video coding for machines [6], in which the subjective quality is much less important than the usability of decoded video for machine vision applications. Therefore, the video can be modified before the compression in order to remove some information redundant for a specific task, decreasing the size of the encoded bitstream. Other examples include changing the representation of geometry before compressing it with JPEG to decrease the amount of data used in GPU-based virtual view synthesis [7], deep preprocessing performed before video encoding [8], changing the projection type of omnidirectional

Manuscript received October 17, 2022. The work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory). (*Corresponding author: Adrian Dziembowski*).

Adrian Dziembowski and Dawid Mieloch are with Poznań University of Technology, 60-965 Poznań, Poland (e-mail: [adrian.dziembowski; dawid. mieloch]@put.poznan.pl). Jun Young Jeong and Gwangsoon Lee are with Electronics and Telecommunications Research Institute, Daejeon, 34129 Republic of Korea (e-mail: [jyj0120; gslee]@etri.re.kr).

video [9], and MPEG-5 part 2 LCEVC (Low Complexity Enhancement Video Coding [10]), which can be used to extend the compression capabilities of a standard video codec by extraction of additional enhancement layer from encoded video. While these approaches concern relatively wide topics related to video compression and should not be directly compared to each other, they follow the idea of performing some kind of processing to improve the efficiency of widespread video codecs. In result, it allows for much faster development of new coding methods, as used internal codecs are often already available in hardware implementations.

The MPEG Immersive Video (MIV) coding standard [11], on which the main focus is put in this paper, follows this abovementioned scheme of video preprocessing. This codec allows efficient encoding for different use cases, including simple free navigation and free viewpoint television systems [12], [13], and more sophisticated virtual reality systems, where a user equipped with an HMD virtually immerses into the captured scene.

In general, MIV decreases the inter-view redundancy and transforms the multiview video data into a much smaller number of videos (called "atlases", Fig. 1) which are later encoded using a typical 2D video encoder. MIV is codec-agnostic, thus the atlases may be encoded using any video encoder, such as HEVC [3] or VVC [4]. More details of the MIV are presented in Section II.

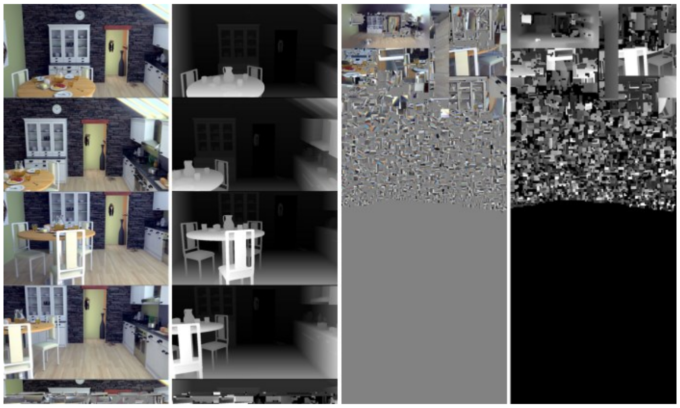

Fig. 1. The output of the MIV encoder: 2 texture and 2 geometry atlases; sequence *Kitchen* [14].

Besides the removal of inter-view redundancy, the MIV encoder already adapts the atlas videos to the characteristics of the widely adopted video encoders using two methods proposed in this paper. While the typical video encoder tries to optimize a bitrate by preserving the highest possible perceived visual quality, it is not optimal strategy for depth map encoding. This assumption can lead to significant degradation of compressed depth maps usefulness in terms of their usability to properly represent the geometry of encoded three-dimensional scenes. Therefore, the first proposed method changes the dynamic range of the geometry (depth) atlas, depending on the quality of input depth maps, in order to decrease the influence of video encoding on the quality of final rendered views. More details about the rationale after this proposal are presented in Section III-A

The second proposed method, patch average color modification, removes the constant component of all YCbCr components of each patch within atlases to decrease the number and magnitude of edges within a texture atlas video. Because of these numerous additional edges between neighboring patches, and between patches and unoccupied areas (see Fig. 1), the distribution of energy in the frequency domain is heavily changed, making their efficient compression much harder when using traditional video encoders. Further details on the rationale are presented in Section IV-A.

The full description of two proposed novel methods is presented in Sections III-B and IV-B. To prove their high effectiveness in the improvement of immersive video compression, their extensive experimental evaluation was performed using the methodology described in Section V. The results of conducted tests are presented in Section VI.

II. MPEG Immersive Video

The development of the MPEG Immersive Video (MIV) started in 2019, and in 2022 it became an official standard [15] thanks to the efforts of experts of the ISO/IEC JTC1/SC29 WG04 MPEG Video Coding group.

A. *MIV Encoding*

As presented in Fig. 2, an MIV encoder processes n input views with corresponding depth maps and camera parameters, and outputs m texture atlases, k geometry (depth) atlases, and metadata which supports descriptions for properly interpreting an MIV bitstream at the decoder side. Typically, $m + k = 4$, by considering the practical hardware constraint of having up to 4 video decoder instantiations [16], [17]. MIV specification [15] defines several profiles adapted for various use cases [18], e.g., MIV Geometry Absent profile [19] dedicated for low-bitrate systems with powerful decoders [20]. However, in this paper, we will focus only on the MIV Main profile, for which $m = k = 2$.

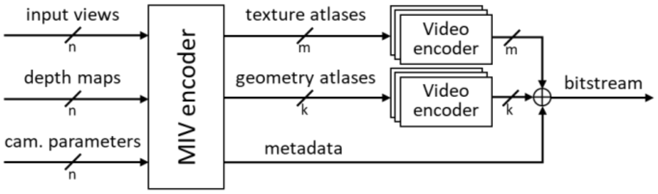

Fig. 2. Simplified scheme of an MIV encoder; typically, for the MIV Main profile, m = 2 and k = 2.

The encoding process in MIV working under the MIV Main profile can be divided into four main steps (Fig. 3): (1) view labeling, (2) pixel pruning, (3) atlas packing, and (4) atlas postprocessing.

In the first step, all input views are analyzed to choose the ones, which carry the most non-redundant information. These views (called "basic views") are inserted into atlases with their entirety. In contrast, all remaining views ("additional views") are further processed to reduce the inter-view redundancy.

Then, in the step of pixel pruning, all inter-view redundant pixels from additional views are removed by cross-projection of pixels between views. After this step, additional views contain only the unique information that is not visible in other views.

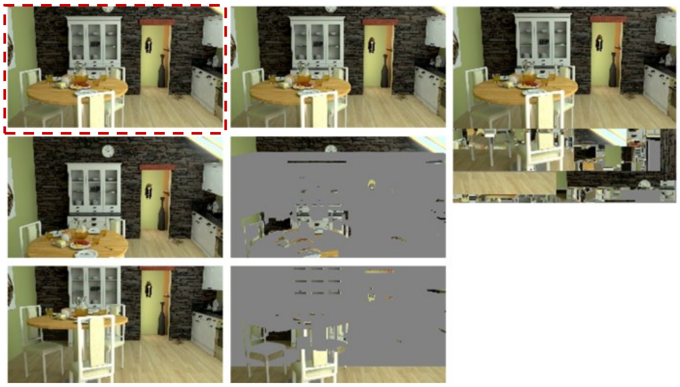

Fig. 3. MIV encoding pipeline. Left: input views (with a dashed outline: basic view); center: views after pruning (only non-redundant areas are preserved); right (all preserved areas packed into the atlas).

In the third step, the preserved (non-pruned) information from all input views is reshuffled and packed into the atlases as a mosaic of patches (cf. second texture and depth atlases in Fig. 1), and the original position of each patch is transmitted within a metadata sub-bitstream (Fig. 2).

In the last step of the MIV encoding, texture and geometry atlases are post-processed to make the atlases "more easily encodable" by a typical video encoder. The postprocessing includes decreasing the resolution of geometry atlases [21], adaptation to the block structure of the video encoder [22], and two techniques that have been proposed by the authors of this paper in internal MPEG documents [23], [24] and are described in two following sections: geometry dynamic range scaling (Section 3) and patch average color modification (Section 4).

A more detailed description of the MIV encoding process can be found in [11], [18], or [25].

B. MIV Decoding

At the decoder side, the received bitstream is demultiplexed into $m + k$ video sub-bitstreams ($2 + 2$ for MIV Main profile) and a single metadata sub-bitstream. Video sub-bitstreams are then decoded using typical 2D video decoders (e.g., VVC), and decoded videos together with metadata are inputted into the MIV decoder (Fig. 4).

The MIV decoder unpacks the atlases and restores input views and corresponding depth maps. Then, these data and camera parameters are inputted to the renderer, which communicates with the final user of the immersive video system. The user signals his or her position and orientation (e.g., by changing the position of the head wearing an HMD device or manipulating a controller when watching the immersive video on a classic 2D display) and the renderer creates demanded viewport.

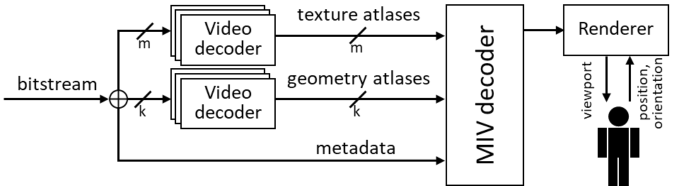

Fig. 4. Simplified scheme of an MIV decoder.

III. Geometry Dynamic Range Scaling

A. Problems of Geometry Atlas Encoding

The purpose of all typical 2D video encoders is to efficiently encode videos captured by a camera, i.e., a texture video. Therefore, it is challenging to efficiently compress a video of a completely different type and characteristics.

The geometry atlas (or, more generally, any depth map video) is usually characterized by two features, which differentiate it from a typical video: sharp edges (no natural blur caused by light capturing [26]), and smooth areas between them (no textures on objects).

The typical video encoder tries to optimize a bitrate by preserving the highest possible perceived visual quality. However, for depth map encoding, this assumption can lead to significant degradation of compressed depth maps usefulness. First of all, the lossy compression negatively influences the accuracy of depth maps, understood as the error in the mapping of the geometry of a three-dimensional scene [27]. Besides decreasing the accuracy of depth maps, the more important in the context of immersive video is the situation in which an edge between two objects is destroyed by blurring and previously non-existent depth values are introduced. These erroneous values result in the wrong reprojection of pixels, thus noticeable artifacts are presented to the user [28].

The first methods for depth map refinement that focused mainly on the reduction of compression-related artifacts were shown relatively early and considered in 3D-TV systems [29]. Nevertheless, the development of new immersive video systems induced further continuous works in this field [30]. Unfortunately, using any post-processing method to decrease the influence of compression on the virtual view synthesis is linked with the increased computational complexity of the decoder. Even for fast deep-learning-based methods, the time required to refine a single frame of HD video is much larger than 40 ms [31], [32], making them very difficult to be used for the real-time implementations of immersive video decoders.

Regarding the fact that the cornerstone of the MIV is to be codec-agnostic, changing the behavior of the video encoder is also not possible and the same encoder has to be used both for coding of texture and geometry atlases.

B. Idea of the Proposed Method

To overcome the abovementioned issues, we have proposed a method that neither modifies the video encoder, nor adds a postprocessing step in the decoder, but makes the geometry atlas easier to be properly encoded with typical video compression methods. Our proposal is based on the general observation, that if the depth difference between two neighboring objects is very small, the magnitude of the edge within a depth map (and geometry atlas) will also be low. In that case, a typical DCT-based video encoder will try to "optimize" (i.e., remove or blur) it, if it will be considered as barely being noticeable to a theoretical viewer, resulting in the appearance of "ghost artifacts" in the synthesized viewport [26]. If the magnitude of an edge is high, an encoder will try to preserve it, but still introducing other artifacts in final viewports (e.g., ringing artifacts), nevertheless, these artifacts are much less visible to the viewer [33].

The proposed method in basics is similar to the luma mapping (LM) tool of Versatile Video Coding [34], which can be used to remap the narrow luma values from the input video to a full range of luma codes, allowing to encode videos more efficiently. However, as discussed earlier, MIV is codec-agnostic, so to use LM only for depth atlas, it would be required to include MIV parser in VVC, breaking the MIV principles in the result. Moreover, the gain from using the scaling would not be available for codecs other than VVC.

C. *Adaptation to Depth Quality*

If the depth map has a good quality (e.g., it was computer-generated using the ground-truth three-dimensional model, or it was measured using a time-of-flight camera), any changes introduced by the video encoder will decrease the quality of the viewport presented to the user. On the other hand, if the depth map was algorithmically estimated on the basis of input views (using any depth estimation software, such as MPEG's IVDE [35], DERS [36], etc.), a slight change of depth value at the edge will have no significant influence on the final quality, but it can significantly decrease the total bitrate (as the edges will be "optimized" by the video encoder).

Taking these observations into account, we have proposed to modify the dynamic range of the geometry video in order to increase or decrease the magnitude of depth edges, depending on the quality of the depth map. The decision about the depth quality is based on the existing automatic depth quality assessment algorithm implemented in the MIV encoder [25] which cross-projects pixels between views for checking the inter-view consistency of depth maps, and decides if the quality of input depth maps is good or not. This automatic assessment is applied on the first frame, where input views are reprojected to the position of the other views. If the reprojected geometry value is higher than the geometry value of the collocated pixel or its neighbors, it is counted as inconsistent, and the quality of the geometry is set to low. A default threshold of 0.1% is used to determine if the inconsistent pixel percentage is too high.

If the MIV depth quality assessor decides that the quality of input depth maps was good, the proposed geometry dynamic range scaling algorithm changes the dynamic range of geometry atlases in order to utilize the full available dynamic range (from 0 to 1023 for 10-bps video, as presented in the center of Fig. 5). If the assessed depth quality is bad, the dynamic range of each geometry atlas is scaled to the half of full available dynamic range (bottom row of Fig. 5).

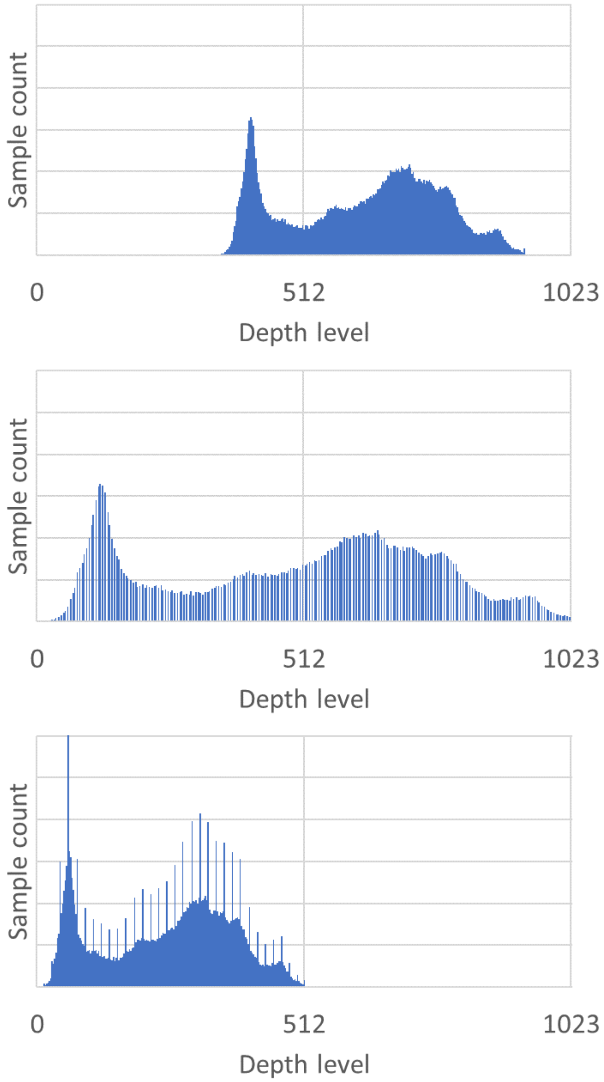

Fig. 5. Idea of proposed geometry dynamic range scaling. Histogram of an original 10-bps geometry video (top); and histogram after modification: if the depth quality is good (center) or bad (bottom).

An example of atlases modified using the proposed approach is presented in Fig. 6. As presented, the magnitude of edges within the modified geometry atlas for content with ground-truth depth (computer-generated *Chess* sequence) was significantly increased, while for natural content with imperfect depth maps (*Painter* sequence), the video encoder received more flexibility to optimize edges, as their magnitude is smaller than without proposed scaling.

Originally, we proposed to modify the geometry dynamic range directly on the geometry atlas [23]. However, the MPEG experts decided to reuse existing syntax elements instead of adding new ones, decreasing the metadata sub-bitstream. Such a decision required to modify a dynamic range of single input views instead of atlases. Therefore, currently the MIV standard specifies, that the dynamic range is changed independently for each input depth map [15]. In most cases, both approaches yield practically the same results as usually the dynamic range of input views is similar among them [11].

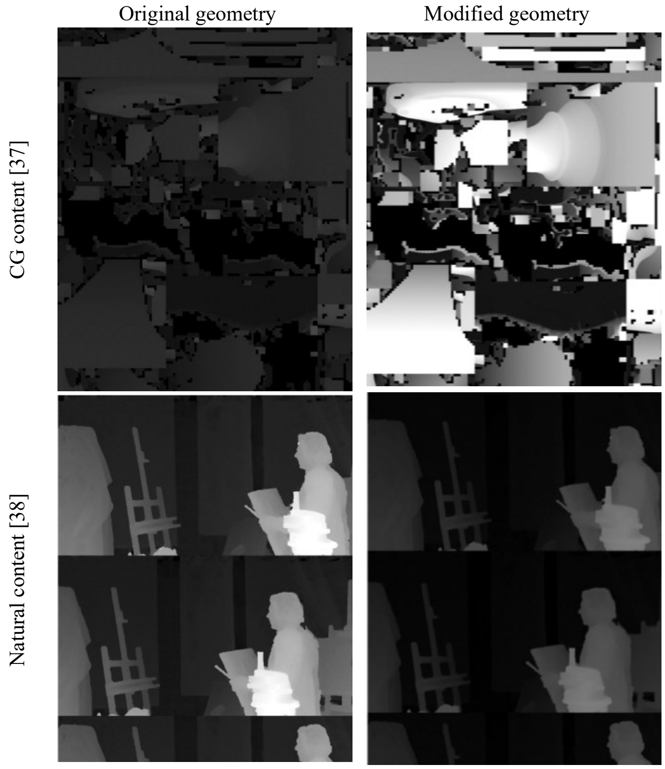

Fig. 6. Fragments of atlases before and after scaling for computer-generated and natural content.

IV. Patch Average Color Modification

A. Problems of Texture Atlas Encoding

As mentioned in the previous section, a typical video encoder is adapted to efficiently process video sequences and to maintain their subjective quality. A texture atlas, regardless of being composed of parts of views acquired by cameras, or computer-generated, significantly differs from the typical video in the aspect of having numerous additional edges between neighboring patches, and between patches and unoccupied areas. It heavily changes the distribution of energy in the frequency domain when compared to traditional videos, making their efficient compression much harder.

These edges are crucial in terms of preserving the high quality of experience. If an edge between two patches will be impaired (e.g., by blurring or introduction of ringing artifacts), the color of pixels at their boundary will be changed. In a typical video, such artifacts can be spotted by the viewer, but they do not heavily influence the subjective quality. In the MIV case, when the decoded atlas is unpacked and the input views are restored, patches with wrongly colored boundaries are moved to their original places. Then, restored input views are used to synthesize viewports presented to the user, resulting in the appearance of clearly visible and disturbing block artifacts (cf. left side of Fig. 16).

Again, taking into account the codec-agnosticism of the MIV encoder, the behavior of a video encoder used to encode the atlases cannot be changed. The earlier-proposed approaches, found in video-based point cloud compression (V-PCC) methods assumed padding of unoccupied pixels to decrease the number of edges between patches and empty spaces [39]. While this approach was shown to be very efficient in these applications, when used in MIV it was already shown to increase the overall bitrate [40], as the characteristics of MIV atlases differ significantly from V-PCC representation. Instead, we propose to modify the texture atlases by decreasing the number of edges between patches and their magnitude.

B. The Proposed Method

The proposed method of patch average color modification changes a constant component (average value) of every patch in texture atlases. Instead of the original value of the constant component, a neutral value is set: $2^{(b-1)}$ for b-bit video (e.g., 512 for 10-bps video; Fig. 7). Hence, instead of filling the empty areas of an atlas, we shift the average color of patches, and as a result, the texture atlases are faded, and the magnitude of edges between patches becomes much smaller (Fig. 8).

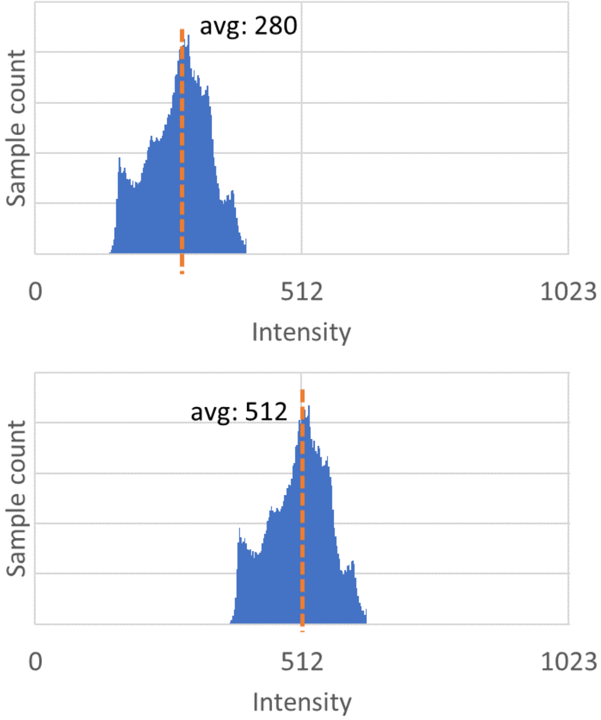

Fig. 7. Idea of proposed patch average color modification. From top: original histogram of color component (e.g., luma) of a patch, and histogram of the patch after modification of the patch average color.

In the case of single-color patches, which occurs often for computer-generated sequences, the proposed modification greys them out, resulting in the complete removal of edges between them and unoccupied atlas space. Fig. 8 also presents an example of such a patch: the large bright horizontal patch at the bottom of the atlas for the *Hijack* sequence.

In order to make the process reversible at the decoder side, the original value of the constant component for luma and both chromas of each patch is transmitted within metadata defined in the MIV specification [15].

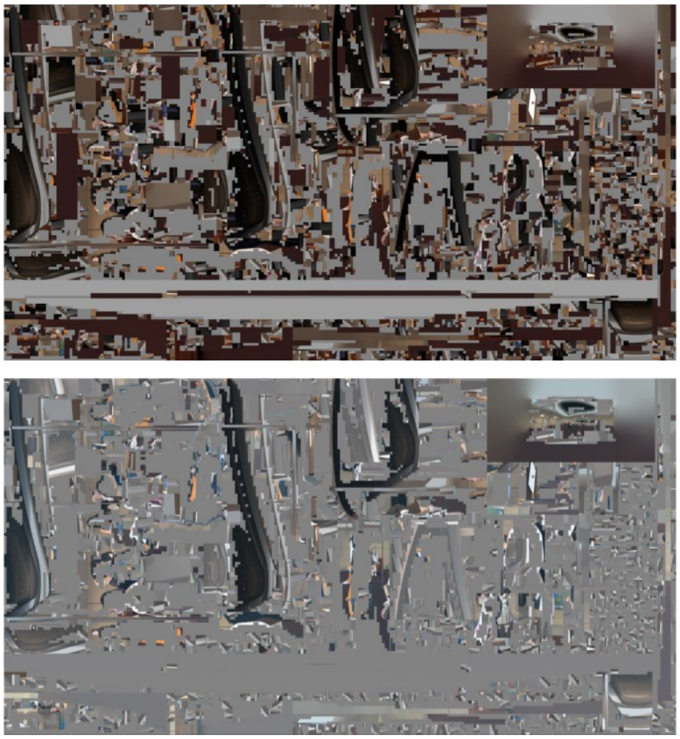

Fig. 8. Atlas before and after patch average color modification; sequence *Hijack* [41].

V. Overview of the Experiments

The performance of the proposed methods is investigated under the robust test conditions established by the MPEG Video Coding group [42]. These conditions are designed for fairly evaluating different competing techniques offered for coding immersive video. TMIV version 13 [25] is applied, but both the encoder and decoder are modified to support the functionalities described in Sections III and IV. For video compression, VVenC, and VVdeC, a fast implementation of VVC is used [43]. Across all the experiment configurations, the following pixel rate constraints are imposed:

- The combined luma sample rate across all decoders shall not exceed 1,069,547,520 samples per second (as in HEVC Main10 profile level 5.2 [44]).
- Each coded video picture size shall not exceed 8,912,896 pixels (i.e., 4096 × 2048).
- The number of decoder instantiations shall not exceed 4.

We use a total of 16 natural and computer-generated multi-view sequences in MIV (Table I). Each sequence has its unique features in terms of camera arrangement (linear, convergent, divergent), input view's resolution (2K, 4K) and projection type (perspective, equirectangular), etc. Across all the experiments, selected 17 frames are used.

TABLE I
List of test sequences.

Sequence	Source	Type		Resolution	Views
ClassroomVideo	[45]	ERP	CG	4096 × 2048	15
Chess	[37]	ERP	CG	2048 × 2048	10
ChessPieces	[46]	ERP	CG	2048 × 2048	10
Hijack	[41]	ERP	CG	4096 × 2048	10
Museum	[41]	ERP	CG	2048 × 2048	24
Group	[47]	Perspective, convergent	CG	1920 × 1080	21
Fencing	[48]	Perspective, convergent	NC	1920 × 1080	10
Fan	[49]	Perspective, planar	CG	1920 × 1080	15
Kitchen	[14]	Perspective, planar	CG	1920 × 1080	25
Cadillac	[50]	Perspective, planar	NC	1920 × 1080	15
Mirror	[51]	Perspective, planar	NC	1920 × 1080	15
Carpark	[52]	Perspective, planar	NC	1920 × 1088	9
Frog	[53]	Perspective, planar	NC	1920 × 1080	13
Hall	[52]	Perspective, planar	NC	1920 × 1088	9
Street	[52]	Perspective, planar	NC	1920 × 1088	9
Painter	[38]	Perspective, planar	NC	2048 × 1088	16

ERP – Equirectangular Projection, CG – Computer-Generated, NC – Natural Content

For performance evaluation, two metrics are used: WS-PSNR [54] and IV-PSNR [55]. Both metrics work in the manner of comparing the acquired ground-truth data existing at the input view's position and the reconstructed data at the same position through view synthesis. Then Bjoentegaard delta (which shows the percentage change in the bitrate required to achieve the same quality for two relevant coding techniques) [56] is calculated for each metric based on four different rate points. The BD-rate values presented in the results section are the averaged ones for all views and then for all test sequences.

VI. Experimental Results

A. Geometry Atlas Modification

In the first experiment, two configurations of the geometry dynamic range scaling were tested and compared with the approach with no dynamic range scaling. In the first configuration, all sequences were scaled to half of the dynamic range ([0, 511] for 10-bps video), while in the second – to the full range [0, 1023].

Obtained results were averaged and presented as RD-curves for two types of content: natural sequences (i.e., sequences with algorithmically estimated depth maps, Fig. 10) and computer-generated sequences with ground-truth depth information (Fig. 9). In order to show the data presented in Figs. 9 and 10 numerically, for each approach two BD-rates were calculated and gathered in Table II. These results, presented separately for each test sequence, can be found also in the final subsection of experimental results (Table VII).

TABLE II
BD-RATES FOR TWO TYPES OF CONTENT; A NEGATIVE VALUE INDICATES INCREASED EFFICIENCY (WHEN COMPARED TO THE NO SCALING APPROACH).

Content type	Half range scaling		Full range scaling	
	IV-PSNR BD-rate	WS-PSNR BD-rate	IV-PSNR BD-rate	WS-PSNR BD-rate
Natural	**– 11.1 %**	**– 15.2 %**	+ 12.4 %	+ 6.7 %
CG	+ 39.0 %	+ 47.4 %	**– 24.8 %**	**– 31.9 %**

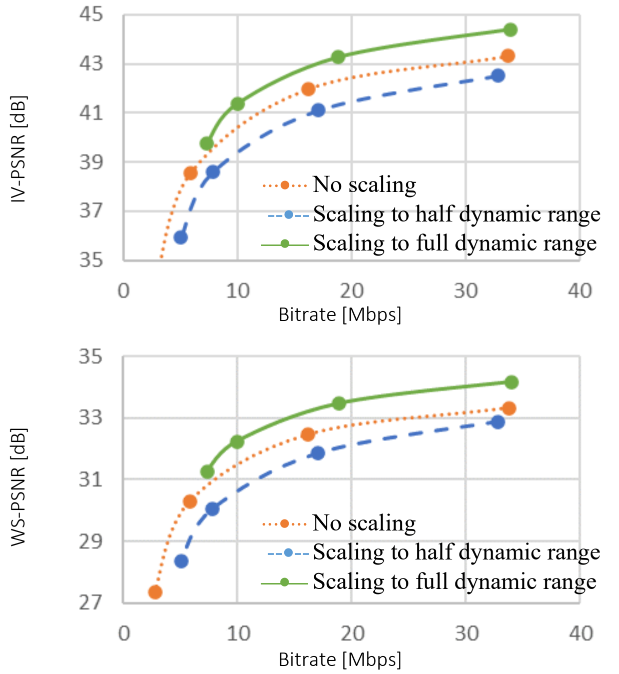

Fig. 9. Comparison of geometry scaling to half and full dynamic range for computer-generated content.

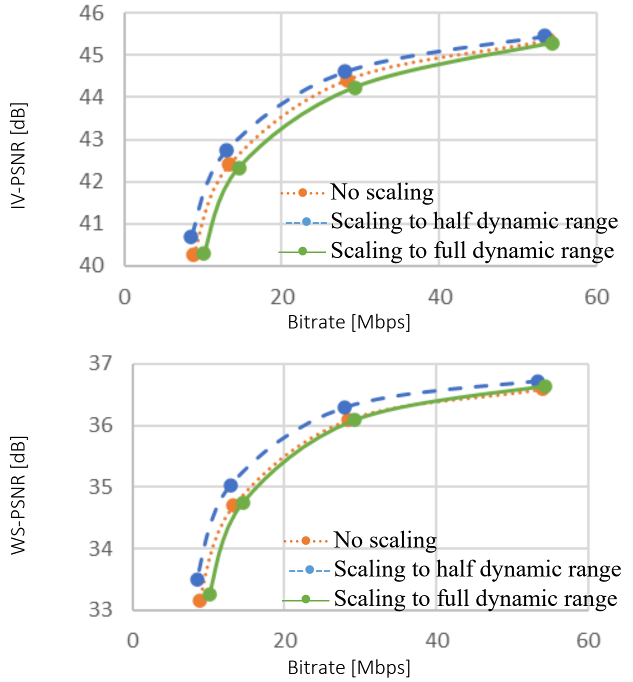

Fig. 10. Comparison of geometry scaling to half and full dynamic range for natural content.

The presented results prove that it is beneficial to scale the dynamic range of the geometry atlases differently depending on the content (i.e., on the input depth quality). In such an approach, the BD-rates averaged over all test sequences are 14.8% and -21.0%, for IV-PSNR and WS-PSNR, accordingly.

In order to present that both methods presented in the paper were adapted specifically to data they were created for, the second experiment was conducted. In this experiment, the geometry scaling method was compared to the approach, where a geometry atlas is processed using the patch average value modification (performed here on depth instead of texture). The results of this experiment are presented in Figs. 11, 12, 13, and Table III.

The RD-curves presented in Fig. 13 show, that both methods can be efficiently used on the geometry atlas, outperforming the approach with no additional processing. However, when comparing the subjective quality of synthesized viewports, the proposed geometry scaling approach provides better subjective quality, both for computer-generated (*Fan*, Fig. 12) and natural sequences (*Carpark*, Fig. 11).

TABLE III
BD-RATES FOR TWO TESTED APPROACHES; A NEGATIVE VALUE INDICATES INCREASED EFFICIENCY (WHEN COMPARED TO THE APPROACH WITH NO GEOMETRY ATLAS PROCESSING).

Proposed geometry scaling		Patch depth average value modification	
IV-PSNR BD-rate	WS-PSNR BD-rate	IV-PSNR BD-rate	WS-PSNR BD-rate
– 14.8 %	**– 21.0 %**	– 8.7 %	– 15.3 %

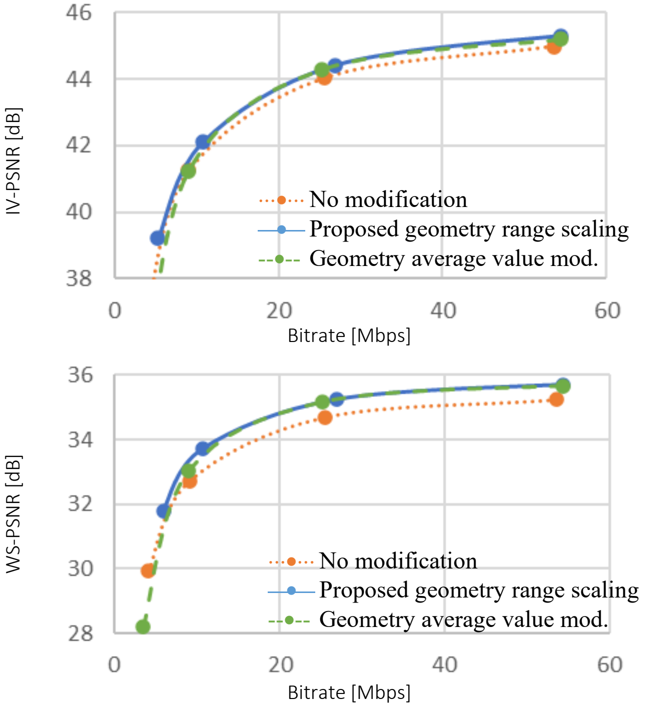

Fig. 13. Comparison of 3 tested approaches.

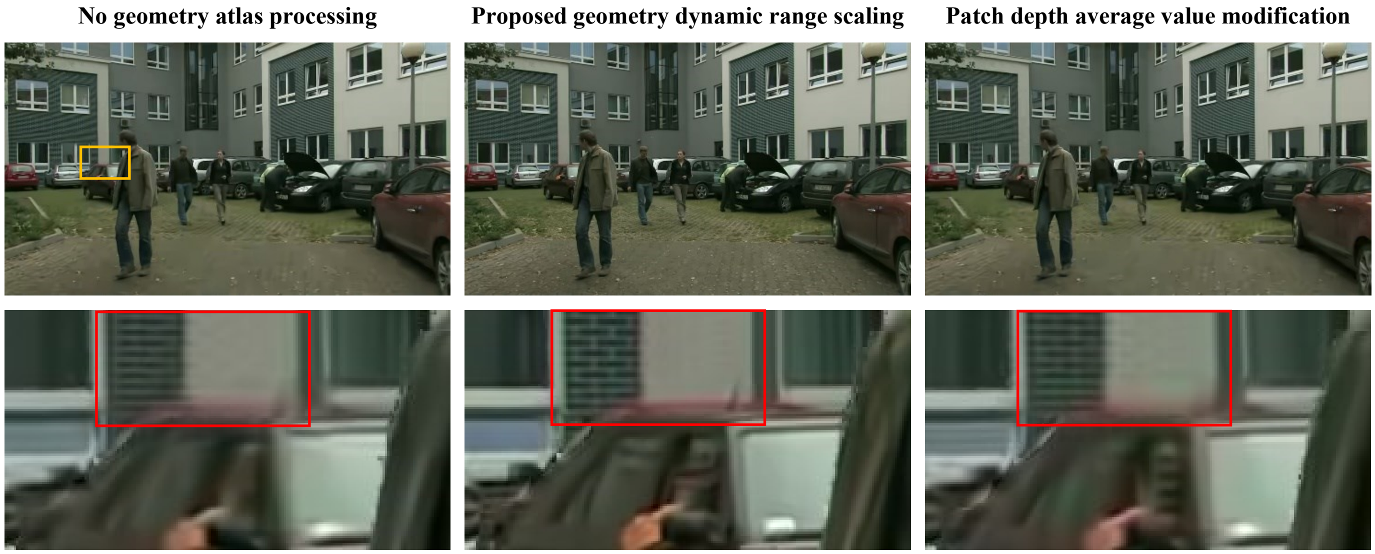
No geometry atlas processing
Proposed geometry dynamic range scaling
Patch depth average value modification

Fig. 11. A viewport (and its fragment) synthesized using atlases processed in 3 ways; left: no geometry atlas processing, center: proposed (geometry dynamic range scaling), right: patch depth average value modification; natural content – sequence *Carpark* [52].

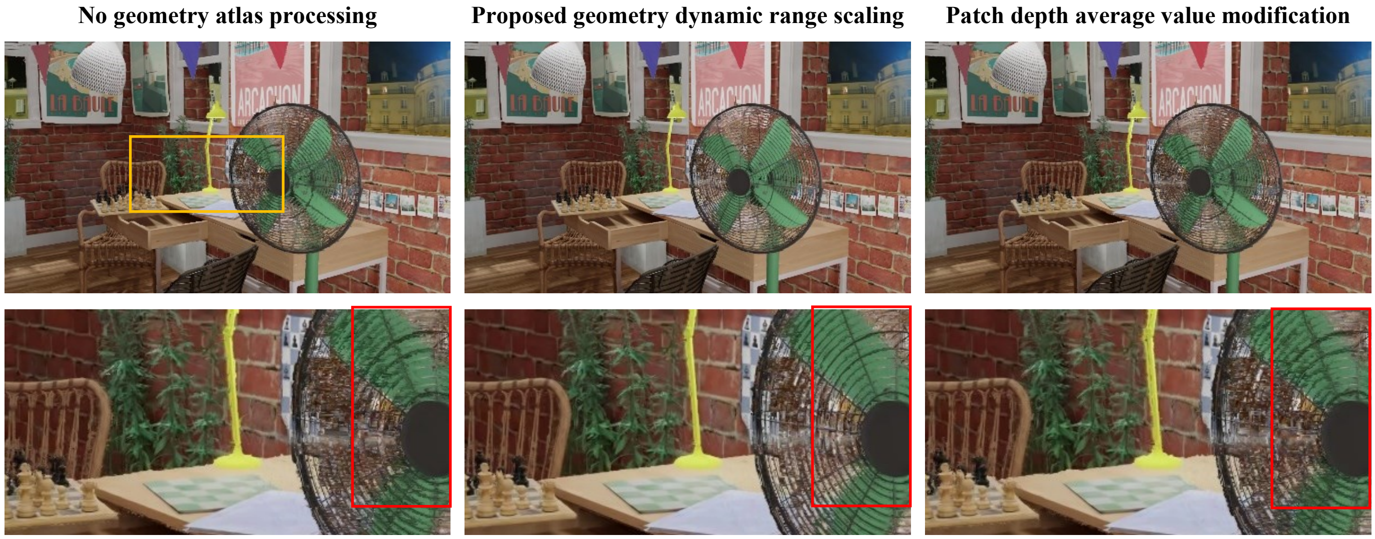
No geometry atlas processing
Proposed geometry dynamic range scaling
Patch depth average value modification

Fig. 12. A viewport (and its fragment) synthesized using atlases processed in 3 ways; left: no geometry atlas processing, center: proposed (geometry dynamic range scaling), right: patch depth average value modification; computer-generated sequence *Fan* [49].

For natural content, where only half of the geometry dynamic range is used, the geometry atlases need fewer bits to be encoded. This is very advantageous for this type of content, as compressed depth maps of natural content tend to be very hard to be efficiently encoded using video encoders. Therefore, when the proposal is used, less-quantized textures can be transmitted using the same total bitrate (e.g., more details on the bricks or car's antenna in the red region of Fig. 11). On the other hand, for computer-generated content, for which the depth maps are usually much easier to encode, the proposed scaling allows for the better preservation of fine edges of objects (e.g., on a fan's grille in the red region of Fig. 12).

B. Texture Atlas Modification

In the third experiment, the efficiency of the proposed patch average color modification method was evaluated. Similar to the previous experiment, the proposed approach was compared to previously shown method but used on a different type of data: dynamic range scaling of the texture atlas. The results are presented in Fig. 14 and Table IV. Results of the patch average value modification, presented separately for each test sequence, can be found also in the final subsection of experimental results (Table VII).

TABLE IV
BD-RATES FOR TWO TESTED APPROACHES; A NEGATIVE VALUE INDICATES INCREASED EFFICIENCY (WHEN COMPARED TO THE APPROACH WITH NO TEXTURE ATLAS PROCESSING).

Proposed patch average value modification		Luminance scaling	
IV-PSNR BD-rate	WS-PSNR BD-rate	IV-PSNR BD-rate	WS-PSNR BD-rate
– 1.3 %	**– 0.6 %**	+ 4.3 %	+ 0.7 %

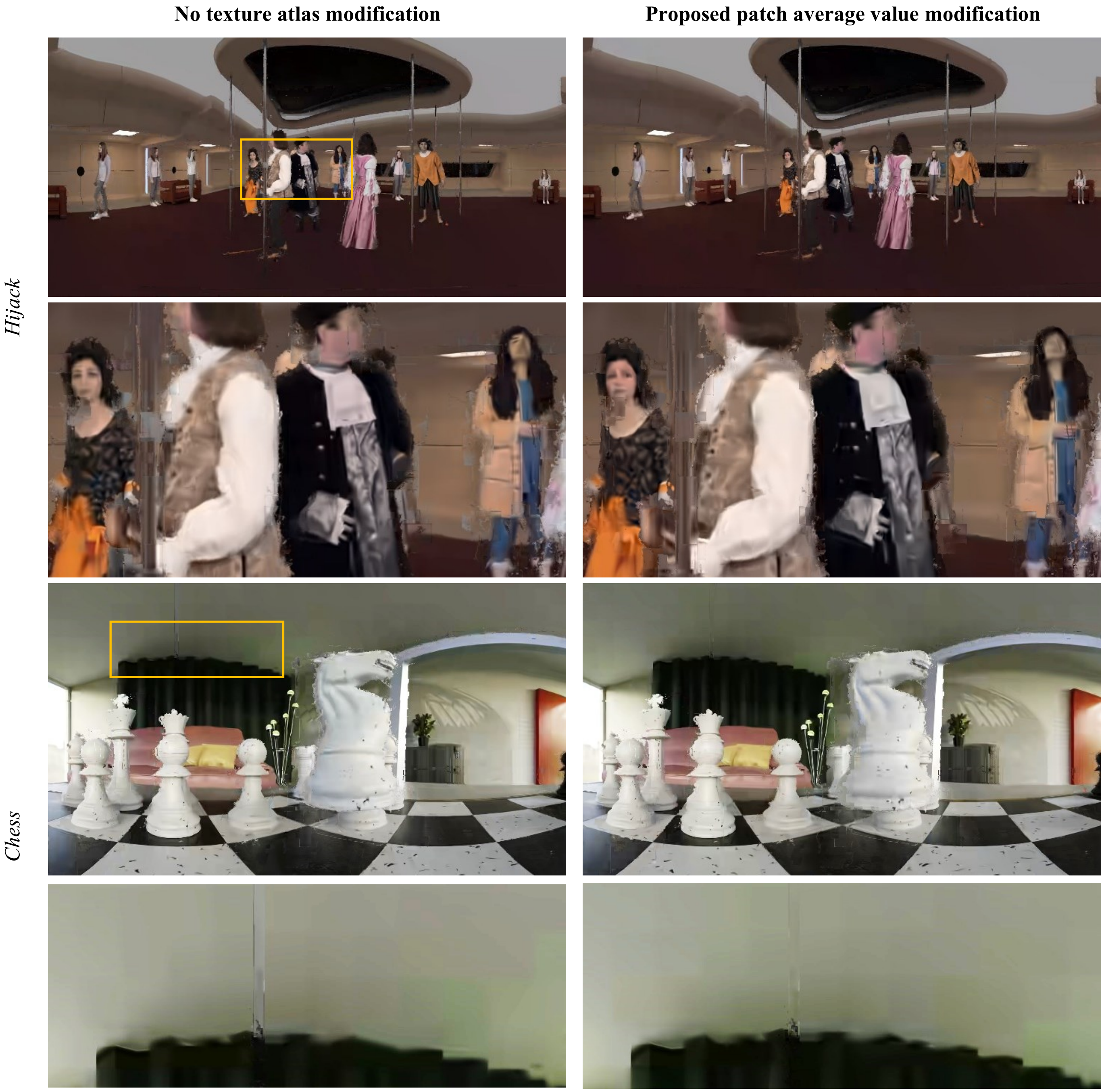
No texture atlas modification
Proposed patch average value modification
Hijack
Chess

Fig. 16. A viewport (and its fragment) synthesized at a very low bitrate; many occlusions (*Hijack* [41]) and smooth area (*Chess* [37]).

Though the objective results presented in Fig. 14 and Table IV show a much smaller change than in the previous experiment, the subjective quality of viewports synthesized using atlases modified with the proposed patch average modification is significantly better than for the approach with no processing of the texture atlases (Fig. 16).

The reason for the discrepancy between subjective and objective results can be explained by the size of areas affected by the proposal, as areas with the wrong color are relatively small, and most of the area of the viewport has no color artifacts. However, the appearance of relatively small grey blocks instead of proper texture, or artifacts on one-color objects like a floor or wall, can be easily spotted by the viewer. Moreover, as presented in Fig. 15, such color artifacts may flicker in time.

The results show also that the luminance scaling decreases the coding efficiency, proving the usefulness of using range scaling for geometry atlases only.

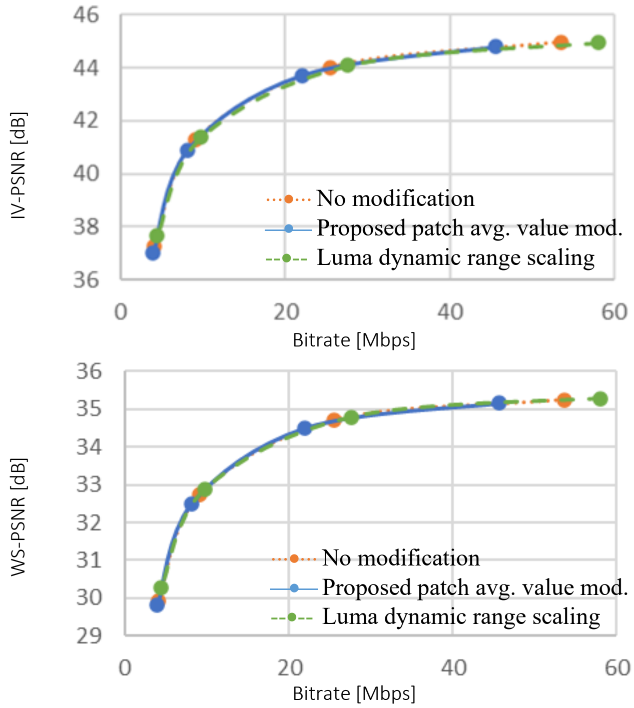

Fig. 14. Comparison of 3 tested approaches.

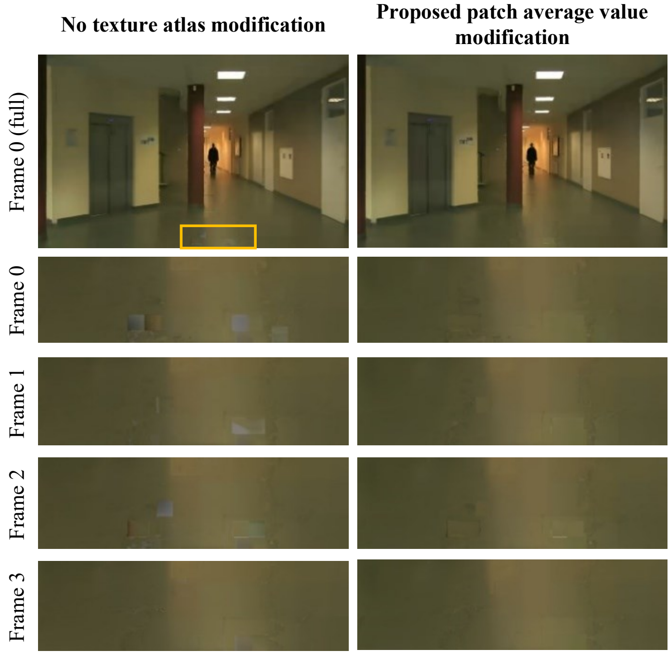
No texture atlas modification
Proposed patch average value modification
Frame 0 (full)
Frame 0
Frame 1
Frame 2
Frame 3

Fig. 15. Fragment of 4 consecutive frames synthesized at very low bitrate; left: without patch average color modification; right: proposed; sequence *Hall* [52].

Table V shows the reduction of the bitrate allocated for texture atlases and the total reduction of bitrate (which also includes the increased size of metadata). It can be seen that for very low bitrates, the proposal significantly decreases the size of encoded texture atlases (up to 10 %), what combined with a very small size of additional metadata (0.05 Mbps), results in up to 6 % of reduction of the overall bitrate). It should be noted that the additional metadata related to the patch average value modification are not quantized, so the size of the metadata could be further reduced.

TABLE V
BITRATE REDUCTION DEPENDING ON THE TOTAL SIZE OF THE MIV BITSTREAM, AVERAGED OVER ALL SEQUENCES.

Total size of the MIV bitstream	Texture atlases bitrate reduction		Total bitrate reduction (incl. metadata)	
50 Mbps	0.56 Mbps	1.7 %	0.51 Mbps	1.5 %
25 Mbps	0.37 Mbps	2.4 %	0.32 Mbps	2.1 %
12 Mbps	0.20 Mbps	2.9 %	0.15 Mbps	2.2 %
7.0 Mbps	0.15 Mbps	5.1 %	0.10 Mbps	3.3 %
3.5 Mbps	0.13 Mbps	9.8 %	0.08 Mbps	6.0 %

C. *Combination of Both Proposed Approaches*

In the last experiment, it was tested how efficient is to combine both proposed approaches. The BD-rates are presented in Tables VI (average values) and VII (per sequence).

The last results show that the combination of both approaches provides a significant reduction of bitrate required to achieve the same quality of the virtual view. Moreover, it should be emphasized that each of the proposed techniques works independently, on different data. Therefore, subjective quality gains caused by both techniques persist after they are combined (Fig. 17).

TABLE VI
BD-RATES FOR THREE TESTED APPROACHES; A NEGATIVE VALUE INDICATES INCREASED EFFICIENCY (COMPARED TO NO ATLAS PROCESSING APPROACH).

Geometry dynamic range scaling		Patch average value modification		Combination of both approaches	
IV-PSNR BD-rate	WS-PSNR BD-rate	IV-PSNR BD-rate	WS-PSNR BD-rate	IV-PSNR BD-rate	WS-PSNR BD-rate
– 14.8 %	– 21.0 %	– 1.3 %	– 0.6 %	**– 15.5 %**	**– 21.2 %**

TABLE VII
WS-PSNR BD-RATES FOR ALL SEQUENCES; A NEGATIVE VALUE INDICATES INCREASED EFFICIENCY (WHEN COMPARED TO THE APPROACH WITH NO ATLAS PROCESSING). "---" INDICATES THAT THE RD-CURVES DO NOT OVERLAP, AND EFFICIENCY OF PROPOSED METHOD IS HIGHER. SEQUENCES HIGHLIGHTED IN ORANGE: COMPUTER-GENERATED, IN BLUE: NATURAL CONTENT.

Sequence	Half range scaling	Full range scaling	**Adaptive scaling**	**Patch avg. value mod.**	**Both proposed approaches**
ClassroomV	6.5%	3.3%	3.3%	-0.7%	2.7%
Museum	-7.5%	-14.0%	-14.0%	-0.3%	-12.0%
Fan	-6.2%	-27.6%	-27.6%	-0.9%	-23.3%
Kitchen	11.3%	-10.5%	-10.5%	0.1%	-10.9%
ChessPieces	33.5%	-7.2%	-7.2%	-1.4%	-8.3%
Cadillac	7.0%	-21.7%	-21.7%	-2.2%	-18.0%
Hijack	-28.5%	-34.6%	-34.6%	-3.2%	-27.8%
Chess	-83.0%	---	---	-1.6%	---
Group	-13.4%	-60.1%	-60.1%	-0.9%	-60.4%
Fencing	-5.4%	16.7%	-5.4%	-0.7%	-6.0%
Street	-5.3%	29.0%	-5.3%	-0.3%	-5.6%
Hall	2.3%	83.3%	2.3%	0.0%	2.4%
Painter	-30.1%	-3.0%	-30.1%	0.2%	-27.0%
Carpark	-16.8%	6.3%	-16.8%	0.0%	-14.4%
Mirror	-13.4%	-3.3%	-13.4%	0.0%	-8.1%
Frog	-4.9%	18.9%	-4.9%	0.7%	-2.0%

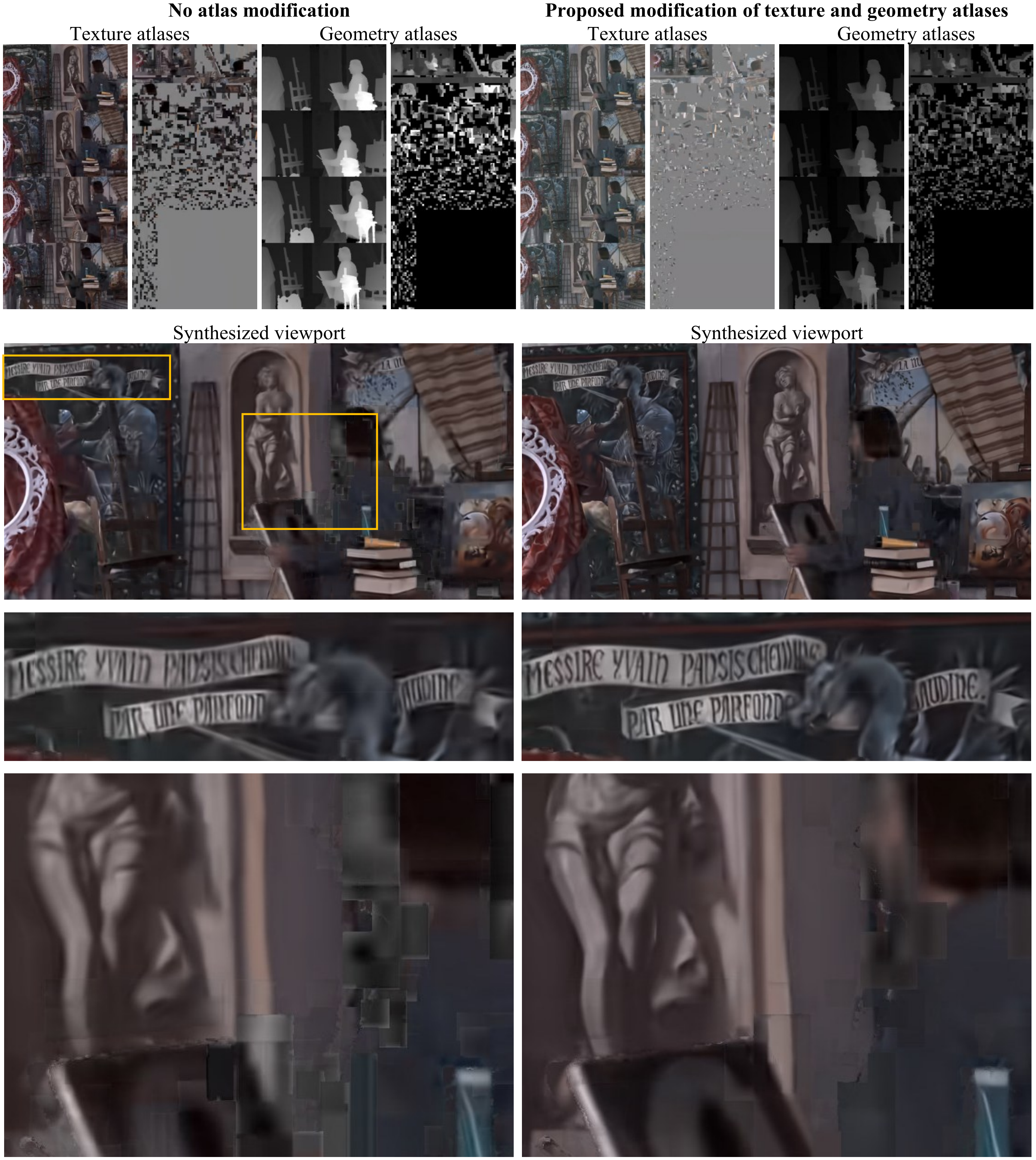

Fig. 17. Combination of both proposed approaches; from top: (1): texture and geometry atlases, (2): synthesized viewport, (3) and (4): fragments of the viewport; sequence *Painter*, very low bitrate: 6 Mbps in both compared cases.

As presented in Fig. 17, combination of both proposed methods significantly reduces color artifacts caused by wrong encoding of patch edges (last row of Fig. 17) and increases the number of texture details in a sequence (3rd row of Fig. 17) due to lower bitrate needed for geometry data.

VII. Conclusions

The paper describes two methods increasing the efficiency of the MPEG Immersive Video (MIV) coding standard [11]. Both proposed methods were appreciated by the experts of the ISO/IEC JTC 1/SC 29/WG 04 MPEG VC group, and are included into the reference software for immersive video coding [25].

The first of the proposed methods modifies the dynamic range of geometry atlases, depending on the quality of input depth maps which is automatically determined within the MIV encoder. For sequences with perfect depth (usually, computer-generated content), the dynamic range of the geometry is expanded to the full range (e.g., [0, 1023] for 10-bps video) to increase the magnitude of edges, making them more insensitive to artifacts introduced by a video encoder. For natural content, where depth maps are estimated and their quality is much lower, the dynamic range of the geometry is set to half of the full range (e.g., [0, 511] for 10-bps video), allowing for decreasing the total bitrate of transmitted immersive video.

In the second proposed method, the texture atlases are modified in order to reduce the magnitude of edges between patches containing non-redundant information from several input views. Such a reduction is possible by the elimination of the constant component of all YCbCr components of each patch. To make the proposed modification reversible at the decoder side, the information about the initial value of the constant components of each patch is sent within metadata defined in the MIV specification [15].

Both proposed methods can work independently or combined, increasing the immersive video coding efficiency even more. Moreover, the fact of being the postprocessing of the atlases generated by the MIV encoder makes both proposed methods possible to be used also together with other methods based on similar principles, e.g., with LCEVC [10], which can be used to further improve the efficiency of used video codec. Studying such a combination would be a natural candidate for further studies, as testing the codec-agnosticism of MPEG immersive video (i.e., if the MIV methods are showing similar efficiency regardless of used video encoder) results in providing further enhancements or use cases of this coding standard [57], [58].

References

[1] M. Wien et al., “Standardization Status of Immersive Video Coding,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 5-17, 2019.

[2] G. Sullivan and T. Wiegand, “Video Compression – From Concepts to the H.264/AVC Standard,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18-31, Jan. 2005.

[3] G. Sullivan et al., “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[4] B. Bross et al. “Overview of the Versatile Video Coding (VVC) Standard and its Applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[5] G. Tech et al., “Overview of the Multiview and 3D Extensions of High Efficiency Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35-49, Jan. 2016.

[6] L. Duan, J. Liu, W. Yang, T. Huang and W. Gao, “Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics,” *IEEE Transactions on Image Processing*, vol. 29, pp. 8680-8695, 2020.

[7] S. Kwak, J. Yun, J.-Y. Jeong, Y. Kim, I. Ihm, W.-S. Cheong, and J. Seo, “View Synthesis with Sparse Light Field for 6DoF Immersive Video,” *ETRI Journal*, vol. 44, pp. 24– 37, Feb. 2022.

[8] A. Chadha and Y. Andreopoulos, “Deep Perceptual Preprocessing for Video Coding,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14847-14856, 2021.

[9] Y. Cai, X. Li, Y. Wang, and R. Wang, “An Overview of Panoramic Video Projection Schemes in the IEEE 1857.9 Standard for Immersive Visual Content Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6400-6413, Sep. 2022.

[10] S. Battista et al., “Overview of the Low Complexity Enhancement Video Coding (LCEVC) Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7983-7995, Nov. 2022.

[11] J. Boyce et al., “MPEG Immersive Video Coding Standard,” *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521-1536, Mar. 2021.

[12] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, “FTV for 3-D Spatial Communication,” *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, Apr. 2012.

[13] O. Stankiewicz et al., “A Free-viewpoint Television System for Horizontal Virtual Navigation,” *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2182-2195, Aug. 2018.

[14] P. Boissonade and J. Jung “Proposition of New Sequences for Windowed-6DoF Experiments on Compression, Synthesis and Depth Estimation,” *ISO/IEC JTC1/SC29/WG11 MPEG2018/M43318*, Ljubljana, July 2018.

[15] ISO/IEC DIS 23090-12, Information technology — Coded Representation of Immersive Media — Part 12: MPEG immersive video.

[16] 2021 TV Video Specifications. Accessed: Jan. 11, 2021. [Online]. Available: https://developer.samsung.com/smarttv/develop/specifications/media-specifications/2021-tv-video-specifications.html

[17] H.264/H.265 Video Codec Unit. Accessed: Jan. 11, 2021. [Online]. Available: https://www.xilinx.com/products/intellectual-property/vvcu.html#overview

[18] V.K.M. Vadakital et al., “The MPEG Immersive Video Standard—Current Status and Future Outlook,” *IEEE MultiMedia*, vol. 29, no. 3, pp. 101-111, Jul.-Sep. 2022.

[19] D. Mieloch et al., “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, Sep. 2022.

[20] 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, May 2022.

[21] B. Sonneveldt and B. Kroon, “Depth-map Scaling for Pixel-rate Reduction,” *ISO/IEC JTC 1/SC 29/WG 11 M52365, Jan. 2020*, Brussels, Belgium.

[22] A. Dziembowski et al. “Spatiotemporal Redundancy Removal in Immersive Video Coding,” *Journal of WSCG*, vol. 30, no. 1-2, pp. 54-62, 2022.

[23] A. Dziembowski et al. “Immersive Video CE1.2: Geometry Scaling,” *ISO/IEC JTC 1/SC 29/WG 11 M54176*, October 2020, Online.

[24] A. Dziembowski et al. “MIV CE3.2: Color-based Patch Analysis,” *ISO/IEC JTC 1/SC 29/WG 11 M54892*, October 2020, Online.

[25] “Test Model 11 for MPEG Immersive Video,” *Doc. ISO/IEC JTC 1/SC 29/WG 04 N 0142*, October 2021, Online.

[26] A. Dziembowski et al., “Multiview Synthesis – Improved View Synthesis for Virtual Navigation,” in *Picture Coding Symposium 2016*, Nuremberg, Germany, 2016.

[27] S. Merrouche, B. Bondžulić, M. Andrić, D. Bujaković, “Accuracy Analysis of Lossless and Lossy Disparity Map Compression,” *Ain Shams Engineering Journal*, vol. 13, no. 3, 2022.

[28] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, T. Wiegand, “The Effects of Multiview Depth Video Compression on Multiview Rendering,” *Signal Processing: Image Communication*, vol. 24, no. 1–2, pp. 73-88, 2009.

[29] D. De Silva, W. Fernando, H. Kodikaraarachchi, S. Worrall and A. Kondoz, "A Depth Map Post-Processing Framework for 3D-TV Systems based on Compression Artifact Analysis," *IEEE Journal of Selected Topics in Signal Processing*, 2011.

[30] M.M. Ibrahim, Q. Liu, R. Khan, J. Yang, E. Adeli, and Y. Yang, "Depth map Artefacts Reduction: A Review," *IET Image Processing*, vol. 14, pp. 2630-2644, Oct. 2020.

[31] X. Wang, P. Zhang, Y. Zhang, L. Ma, S. Kwong, J. Jiang, "Deep Intensity Guidance Based Compression Artifacts Reduction for Depth Map," *Journal of Visual Communication and Image Representation*, vol. 57, pp. 234-242, Nov. 2018.

[32] H. Zhang, Y. Zhang, L. Zhu and W. Lin, "Deep Learning-Based Perceptual Video Quality Enhancement for 3D Synthesized View," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5080-5094, Aug. 2022.

[33] K. Müller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, Apr. 2011.

[34] M. Karczewicz et al., "VVC In-Loop Filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3907-3925, Oct. 2021.

[35] "Manual of Immersive Video Depth Estimation 3," *ISO/IEC JTC 1/SC 29/WG 04 MPEG VC N0058*, Jan. 2021.

[36] S. Rogge et al., "MPEG-I Depth Estimation Reference Software," in *International Conference 3D Immersion (IC3D)*, Dec. 2019.

[37] L. Ilola, V.K.M. Vadakital, K. Roimela, and J. Keränen "New Test Content for Immersive Video – Nokia Chess," *ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787*, Geneva, September 2019.

[38] D. Doyen et al., "Light Field Content from 16-camera Rig," *ISO/IEC JTC1/SC29/WG11 MPEG2017/M40010*, Geneva, January 2017.

[39] L. Li, Z. Li, S. Liu, and H. Li, "Efficient Projected Frame Padding for Video-Based Point Cloud Compression," *IEEE Transactions on Multimedia*, vol. 23, pp. 2806-2819, 2021.

[40] H.H. Kim et al., "[MPEG-I Visual] CE3-related: Atlas Padding," *ISO/IEC JTC 1/SC 29/WG 11 M53698*, April 2020, Online.

[41] R. Doré, "Technicolor 3DoF+ Test Materials," *ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349*, San Diego, March 2018.

[42] "Common Test Conditions for MPEG Immersive Video," *ISO/IEC JTC 1/SC 29/WG04 N0169*, Jan. 2022.

[43] A. Wieckowski et al., "VVenC: An Open And Optimized VVC Encoder Implementation," in *Proceedings IEEE International Conference on Multimedia Expo Workshops (ICMEW),* 2021.

[44] ISO/IEC 23008-2, Information Technology — High Efficiency Coding and Media Delivery in Heterogeneous Environments — Part 2: High Efficiency Video Coding.

[45] B. Kroon, "3DoF+ Test Sequence ClassroomVideo," *ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415,* San Diego, April 2018.

[46] L. Ilola, V.K.M. Vadakital, "[MPEG-I Visual][MIV] Improved NokiaChess Sequence," *ISO/IEC JTC1/SC29/WG11 MPEG2020/ M54382*, Online, July 2020.

[47] R. Doré, G. Briand, and F. Thudor, "InterdigitalGroup Content Proposal," *ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731*, Online, June 2020.

[48] M. Domański et al., "Multiview Test Video Sequences for Free Navigation Exploration Obtained using Paris of Cameras," *ISO/IEC JTC1/SC29/WG11 MPEG2018/M38247*, Geneva, May 2016.

[49] R. Doré, G. Briand, and F. Thudor, "InterdigitalFan Content Proposal for MIV," *ISO/IEC JTC1/SC29/WG11 MPEG/M54732*, Online, June 2020.

[50] R. Doré, G. Briand, and F. Thudor, "[MIV] New Cadillac Content Proposal for Advanced MIV v2 Investigations," *ISO/IEC JTC1/SC29/WG04 MPEG VC/M57186*, Online, July 2021.

[51] R. Doré and G. Briand, "Interdigital Mirror Content Proposal for Advanced MIV Investigations on Reflection," *ISO/IEC JTC1/SC29/WG11 MPEG2020/M55710*, Online, January 2021.

[52] D. Mieloch, A. Dziembowski, and M. Domański, "Natural Outdoor Test Sequences," *ISO/IEC JTC1/SC29/WG11 MPEG2019/M51598*, Brussels, January 2020.

[53] B. Salahieh et al. "Kermit Test Sequence for Windowed 6DoF Activities," *ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748*, Ljubljana, July 2018.

[54] Y. Sun et al., "Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1408-1412, Sep. 2017.

[55] A. Dziembowski et al. "IV-PSNR – The Objective Quality Metric for Immersive Video Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7575-7591, 2022.

[56] G. Bjoentegaard, "Calculation of Average PSNR Differences Between RD-Curves," *ITU-T VCEG Meeting*, Austin, USA, 2001.

[57] J. Samelak, A. Dziembowski, D. Mieloch, "Advanced HEVC Screen Content Coding for MPEG Immersive Video", *Electronics*, vol. 11, no. 23, Dec. 2022.

[58] A. Grzelka, A. Dziembowski, D. Mieloch, M. Domański, "The Study of the Video Encoder Efficiency in Decoder-Side Depth Estimation Applications", in *30th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision : WSCG 2022*, Pilsen, Czech Republic, 17-20 May 2022.

Adrian Dziembowski was born in Poznań, Poland in 1990. He received the M.Sc. and Ph.D. degrees from the Poznan University of Technology in 2014 and 2018, respectively. Since 2019 he is an Assistant Professor at the Institute of Multimedia Telecommunications. He authored and coauthored over 40 articles on various aspects of immersive video, free navigation, and FTV systems. He is also actively involved in ISO/IEC MPEG activities towards MPEG immersive video coding standard.

Dawid Mieloch received his M.Sc. and Ph.D. from Poznań University of Technology in 2014 and 2018, respectively. Currently, he is an assistant professor at the Institute of Multimedia Telecommunications. He is actively involved in ISO/IEC MPEG activities where he contributes to the development of the immersive media technologies. He has been involved in several projects focused on multiview and 3D video processing. His professional interests include also free-viewpoint television, depth estimation and camera calibration.

Jun Young Jeong received his BS and MS degrees in electrical engineering in 2013 and 2016, respectively from Purdue University, West Lafayette, IN, USA. He has been a research staff in the Immersive Media Research Laboratory, ETRI, Rep. of Korea since 2016, and has primarily been involved in the development of camera systems for acquiring immersive 6DoF VR and depth estimation software by using stereo vision algorithms. His current research interests include image processing and computer vision, especially in the field of deep-learning based depth estimation.

Gwangsoon Lee received his PhD degree in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2004. He joined the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, in 2001. He is currently a Principal Researcher with Realistic-Media Research Section. His research interests include immersive video processing, light field imaging system, and three-dimensional video system.