A New Approach to Decoder-Side Depth Estimation in Immersive Video Transmission

Dawid Mieloch, *Member, IEEE*, Adrian Dziembowski, *Member, IEEE*,
Dominika Klóska, Błażej Szydełko, Jun Young Jeong, and Gwangsoon Lee

Abstract— This paper presents a novel approach to decoder-side depth estimation (DSDE) in immersive video transmission. The proposal improves the state-of-the-art coding approach by changing the encoding and decoding processes so that the decoder-side depth estimator can utilize partial geometry information about the scene from several available input depth maps. Proposed changes allow for significantly faster decoding and better quality of virtual viewports presented to the viewer. The paper proposes two approaches, namely input depth map assistance (IDMA) and extended IDMA (eIDMA), both compliant with the bitstream of MPEG immersive video (MIV) standard. IDMA involves sending full depth maps for a selected subset of input views, which are then used to refine and enhance the depth maps for the remaining views. In eIDMA, the decoder additionally reprojects decoded depth maps to the remaining views and enhances them with depth patches containing difficult-to-estimate areas. Proposed methods were tested under the ISO/IEC MPEG Video Coding common test conditions for MIV. The combination of two proposals, the adaptive IDMA, was shown to outperform the current state-of-the-art DSDE approach in the rendered video quality and the computational complexity of decoder. MPEG experts have appreciated the proposed approaches, which will comply with a new DSDE profile of the incoming second edition of the MIV standard.

Index Terms— depth map, immersive video, decoder-side depth estimation, video codecs, video processing

I. Introduction

The immersive video system is a system that allows the user (viewer) to virtually move in a three-dimensional scene recorded using multiple cameras or generated using computer graphics. Virtual navigation can be carried out by head-mounted displays (HMD) [1], traditional monitors with an attached control device, or using touch-sensitive tablets [2], [3].

Choosing a preferable viewpoint by the end user is associated with each viewer independently deciding where they look. Thus, in order to enable the simultaneous transmission of the immersive video to many users, the broadcaster cannot transmit only one specific video stream. Instead, it is necessary to send a vast amount of data consisting of views' texture information from multiple cameras and some representation of the geometry information. The most commonly used representation of an acquired scene is MVD (multiview video plus depth) [4], in which the content and geometry of the scene are stored in the form of multiple views and depth maps.

In the case of such a significant increase in the amount of transferred data in immersive video systems, when compared to traditional television, typical video coders turn out to be insufficiently effective [5]. This is especially noticeable if we need to meet the constraint of pixels that can be decoded per second (*pixel rate*) in nowadays hardware decoders [6] or if the available bandwidth of the existing transmission system is not sufficient. These aspects make it very difficult to create a practical immersive video system without the use of dedicated compression algorithms which, e.g., consider the inter-view redundancy to decrease the size of the encoded bitstream. An overview of such relevant compression methods is included in Section II.

Most of immersive video compression methods assume that depth maps included in a bitstream are compressed using typical (or adapted [7]) video codecs [8]. Using video encoding for depth maps of computer-generated sequences is relatively efficient, as such depth maps do not contain errors related to depth estimation based on the texture of input views and are highly temporally consistent. On the other hand, the compression of depth maps for natural content is much more problematic. When compressed, these depth maps often require a much higher bitrate to still provide a satisfactory quality of virtual view synthesis [9]. When the required bitrate is unavailable, depth maps can contain blocking artifacts, blurring, ringing, and discontinuities, particularly around depth object boundaries. Differently than in the color image, degradation of the depth map will lead to an error in 3D scene reconstruction, causing more noticeable artifacts in the

Manuscript received March 10, 2023, revised June 13, 2023. 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: Dawid Mieloch*).

Dawid Mieloch, Adrian Dziembowski, Dominika Klóska, and Błażej Szydełko are with Poznań University of Technology, 60-965 Poznań, Poland (e-mail: [dawid.mieloch; adrian.dziembowski; dominika.kloska; blazej.szydelko]@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).

synthesized views [7]. Thus, reducing compression artifacts in depth maps used for immersive video applications is crucial.

An efficient depth refinement technique could be considered to improve the efficiency of depth map compression, as improvement of temporal and inter-view consistencies of depth maps can significantly increase the quality of virtual views synthesized for the final user [10]. Such refinement can be performed either as pre-processing before encoding, improving the effectiveness of depth compression [11], or after decoding, to remove compression artifacts in post-processing [12]. Any suitable pre-processing method can improve the quality of depth maps, as this step can be considered part of sequence acquisition and is independent of the encoding method used in the next step. On the other hand, the post-processing method useful for immersive video applications needs to be versatile to handle multiview videos of very diverse characteristics (e.g., types of cameras and their arrangement) but also variable compression errors for different bitrates [5]. Moreover, an additional computationally expensive step increases the time required to render a new virtual view for the user, making real-time implementations much more challenging to provide.

In order to overcome the problems with depth map compression, researchers considered a scheme called decoder-side depth estimation (DSDE) [6]. In such a compression method, the depth estimation step is moved from the process of the scene acquisition, done before encoding, to the decoder and is performed on compressed full views [13], [14]. Based on this coding method, we propose a novel method of immersive video compression with simultaneous multiview depth estimation and refinement performed in the decoder using a set of compressed depth maps included in the bitstream – Input Depth Map Assistance (IDMA).

The proposal changes the essential principle of previous approaches in encoding based on decoder-side depth estimation (described in Section II) that depth maps are not encoded in the bitstream, as they are estimated in the decoder using decoded textures of input views and camera parameters [15]. Our approach assumes that a subset of depth maps acquired in the depth estimation process occurring at the encoder side is also available in the decoder of immersive video. Then, the decoded depth maps available in the decoder are utilized in multiview depth estimation. It enables simultaneous enhancement of their quality, and the quality of depth maps estimated for remaining views, as the set of transmitted depth maps helps the estimator in the reconstruction. Moreover, as depth maps are not sent for all views, we preserve the high share of the texture of views in the bitrate, keeping their quality sufficiently high to provide good-quality depth estimation and virtual view synthesis.

We also propose a further extension of the proposal – extended IDMA (eIDMA). We modify immersive video encoding with a unique feature that allows the reprojection of decoded input depth maps to other views and the subsequent addition of supplemental depth patches (fragments) for further enhancement. An additional set of patches includes the elements of a scene that were not visible in the transmitted neighboring views and could not be estimated from available views. These elements, however, could be recovered from the depth maps of the non-transmitted views existing at the encoder side. Therefore, such elements could be further used to assist in inferring regions that could not be reconstructed by simple depth reprojection from available views.

As discussed earlier, the existing video encoding methods (e.g., HEVC [16] or VVC [17]) are not sufficiently efficient for depth maps. However, the proposed method of modified DSDE, which includes depth refinement, was demonstrated in performed experiments to provide a significantly better quality of decoded virtual views and simultaneously decrease the computational complexity in comparison with other DSDE-based immersive video compression methods.

To summarize, the novelties of the proposals described in this paper are as follows:

- IDMA coding scheme:
  - sends only a subset of depth information, preserving a high share of texture information in the bitrate, ensuring good-quality depth estimation and virtual view synthesis,
  - runs the view labeling algorithm twice to properly select the depth maps to be sent, ensuring maximum scene coverage and minimal overlap between transmitted depth maps,
  - involves utilizing depth maps available in the decoder to improve the quality of depth maps estimated for the rest of views,
  - employs a modified depth estimation method based on global multiview optimization to refine the available input depth maps in the decoder.
- eIDMA coding scheme:
  - includes additional reprojection of available input depth maps to other views, further improving the quality of estimated depth maps,
  - in addition to the full-view depth maps in IDMA, sends the depth atlases that contain partial depth information for the rest of views,
  - feds resulting depth maps into the modified depth estimator, which estimates new depth only for empty regions, reducing computational complexity.
- Adaptive IDMA coding scheme:
  - enhances the encoder with an automatic selection mechanism for determining the most suitable method, dynamically choosing between the eIDMA and IDMA approaches, depending on the characteristics of the content being processed.
  - further improves the efficiency and performance of the encoding process, ensuring that the possibly best results are achieved for both computer-generated and natural content.

After the review of state-of-the-art immersive video compression in Section II, Section III provides a detailed description of the proposal. Section IV explains the methodology of performed experimental tests, describes the used dataset, and, finally, includes the comprehensive comparison of the compression efficiency and the measured runtimes of tested methods. Section V concludes the paper and provides remarks on future works and forecasted applications of the proposal.

II. Immersive Video Compression

A. Basic solutions

MVD simulcast encoding stands for the compression method in which each texture and depth video is encoded independently using the typical 2D video codec (such as AVC [18], HEVC [16]) without considering the spatially redundant information commonly existing at neighboring views [19]. This method has the merit of simple implementation, but it requires high-performance devices that support the simultaneous running of several video decoders and a large memory buffer for storing multiple videos for rendering. For most of the widely-used consumer devices, it is hard to satisfy these requirements, as they typically allow up to 4 simultaneous decoders for 4096 × 2048 videos at 30 fps [20], [21]. Therefore, for immersive video, for which the number of input views significantly exceeds these values, the MVD simulcast cannot be considered to be an optimal solution.

Unlike MVD simulcast, MV-HEVC and 3D-HEVC [22] exploit the similarity information among neighboring views for supporting inter-view prediction so that more efficient compression can be possible. However, these two tools still have restrictions that hinder their broader adoption in immersive video systems. First, they are only designed to work properly with the MVD data captured by the coplanar (or even linear) arrangement of traditional cameras with a narrow baseline, thus, the size and shape of the viewing zone are highly limited in comparison with immersive systems composed of omnidirectional cameras [23]. Moreover, these techniques are built on top of the HEVC codec, therefore, it is impossible to improve its performance by utilizing newer compression standards, such as VVC [17]. In addition, further minor drawbacks, such as the reference views that will be used to predict non-reference views, must be manually determined, causing further usability issues.

B. MPEG immersive video (MIV)

The current state-of-the-art compression technology for immersive video was developed by the ISO/IEC MPEG Video Coding group under the name MPEG immersive video (MIV) [23], [24]. The idea used in the MIV Main profile of this standard is the assumption that a certain small number of basic views, gathering most of the scene information, should be fully encoded, while supplementary information visible from non-basic ("additional") views may be communicated in the form of small fragments ("patches") (Fig. 1), or omitted, enabling increasing the number of basic views.

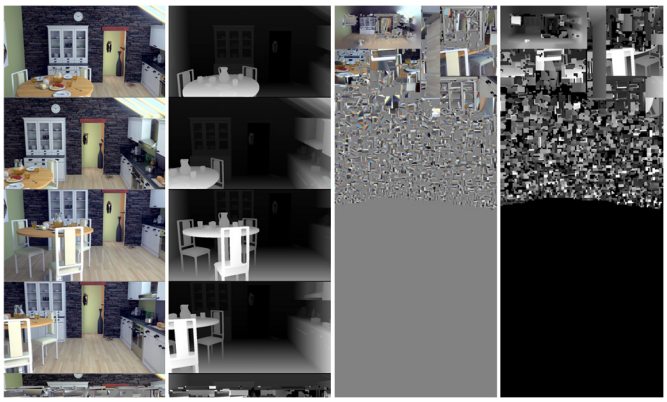


Fig. 1. Example of four atlases produced by MIV encoder running in MIV Main profile: first and third atlas contain texture information in the form of full views and patches, while the second and fourth atlases contain corresponding depth.

Unlike the previous multiview coders, which define the entire encoding process, from reading input views to creating one common bitstream, the MIV encoder (left part of Fig. 2) is a kind of pre-processing of multiview video. The encoder removes the inter-view redundancy during the pixel pruning step and changes the representation of the input data (from separate views to "atlases" containing packed information from many views). The MIV decoder (right part of Fig. 2) acts as post-processing of decoded data, recovers the input views from atlases, and produces a virtual view for the position and orientation requested by the final user.

The MIV atlases can be coded using any 2D video encoder, making the MIV a codec-agnostic technique. Thus, with MIV, it is possible to use effective yet time-consuming compression using the newest VVC encoder, as well as a less effective but much faster HEVC [16].

While this freedom in terms of the used video encoder is an unquestionable advantage of the MIV Main, the standard assumes using the same encoder both for textures and depth maps. The DSDE scheme, introduced in Section I, overcomes the problem of using the video encoder not adapted to properly compress depth maps by the depth estimation moved to the decoder. In the case of MIV, this scheme is implemented in Geometry Absent (GA) profile, which was shown to provide very high efficiency for low bitrates (smaller than 10 Mbit/s for a whole three-dimensional scene) [6].

C. Decoder-side depth estimation (DSDE) in MIV coding

In order to facilitate the decoder-side depth estimation process, which requires inter-view redundancy to estimate depth properly, the pixel pruning is not performed in the encoder for this profile (Fig. 2), therefore, produced atlases contain only full views (Fig. 3). Some first modifications of this profile, proposed in [25], shown also using partially complete views composed of patches for decoder-side depth estimation.

Naturally, DSDE requires the use of high-quality depth estimation, which is already possible in the decoder, but such estimation is usually highly time-consuming [6], [26], making it much harder to use this scheme in real-time applications. The possibility of using additional depth refinement can also be

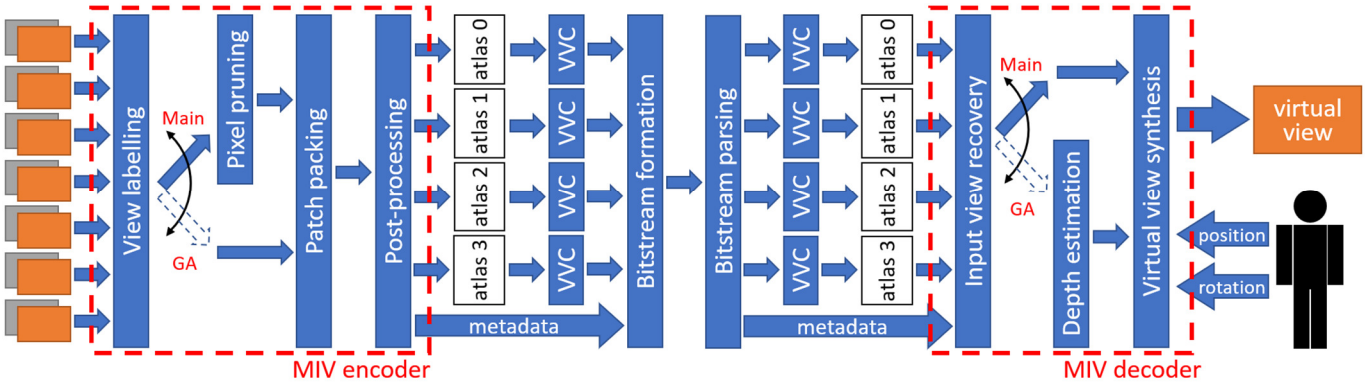


Fig. 2. The simplified scheme of MIV encoder and decoder in MIV Main and MIV Geometry Absent (GA) profiles.

considered, nevertheless, it would further increase the complexity of the decoder.

The quality of depth maps estimated in the decoder is lower than for depth maps estimated before the encoding. First, the number of views available in the decoder, due to pixel-rate constraints, is smaller than in the encoder. The depth maps quality was shown in [10] to depend on the number of views used in the estimation process. Besides that, the quality of input views is lower in DSDE due to compression-induced errors, which also affect the quality of estimated depth [14].

The further important disadvantage of DSDE is not utilizing high-quality depth maps, which usually are available on the encoder side. One of the previously proposed solutions is utilizing a set of encoder-derived features [6], [15], [27], which helps the decoder-side estimator to improve the quality of depth maps and to speed up the process. It is done by including into the bitstream information, e.g., how to narrow the range of possible depth levels which should be considered for each block of the depth map or which block can be skipped in the estimation, as the depth values from the previous frame can be copied and applied for static regions.

Unfortunately, in this scheme, it is not possible to use a depth estimator which is not adapted to utilize these features, highly narrowing the set of suitable methods. Moreover, features can use from 1 to 2 Mbit/s of available bandwidth [27], [28], so for low bitrates (less than 5 Mbit/s), the bitrate left for textures is very low, affecting the delivery of sufficient information for enabling accurate depth estimation and virtual view synthesis. Furthermore, as features are block-based, it is hard to encode more advanced structures/elements from the depth map, as using small blocks increases the bitrate significantly (changing the grid size of 128 × 128 pixels to 64 × 64 doubles the bitrate [27]).

The approach described in [29] employs depth estimation in both the encoder and decoder. The encoder-side estimation leverages motion vectors extracted from encoded texture videos to determine whether the depth map for a given block should be estimated from decoded views (similar to default DSDE) or reconstructed from previous depth maps using motion compensation. This method effectively shifts a substantial portion of the computational burden from the decoder to the encoder, resulting in approximately 20 times faster decoding for the optimal parameter settings. Nevertheless, as the proposal assumes that motion vectors are extracted from videos encoded with AVC encoder, it breaks the assumption that any video codec can be used, so it is not compatible with MIV framework.

Fig. 3. Example of four atlases produced by MIV encoder running in MIV Geometry Absent profile: all four atlases contain full views.

III. Input Depth Map Assistance

A. Overview

This section provides a detailed description of our two proposals for new immersive video compression methods: input depth map assistance (IDMA) and extended input depth map assistance (eIDMA).

The main idea of the proposals is shown in Fig. 4. Both of the proposed methods are based on the scheme of encoding a part of depth maps in order to improve the depth estimation performed at the decoder side, which is usually performed only on the basis of textures of available views (Fig. 4a). In the IDMA approach, the input depth maps, available for a subset of views, are simultaneously refined and used to improve the quality of depth maps estimated for other views (Fig. 4b). It is possible by utilizing the modified depth estimation method based on global multiview optimization.

In order to further improve the quality of estimated depth maps and decrease the computational complexity of their estimation, we propose the eIDMA approach (Fig. 4c), in which the set of available input depth maps is reprojected to other views for which the input depth maps are not available for all pixels. These reprojected depth maps are also fed into the

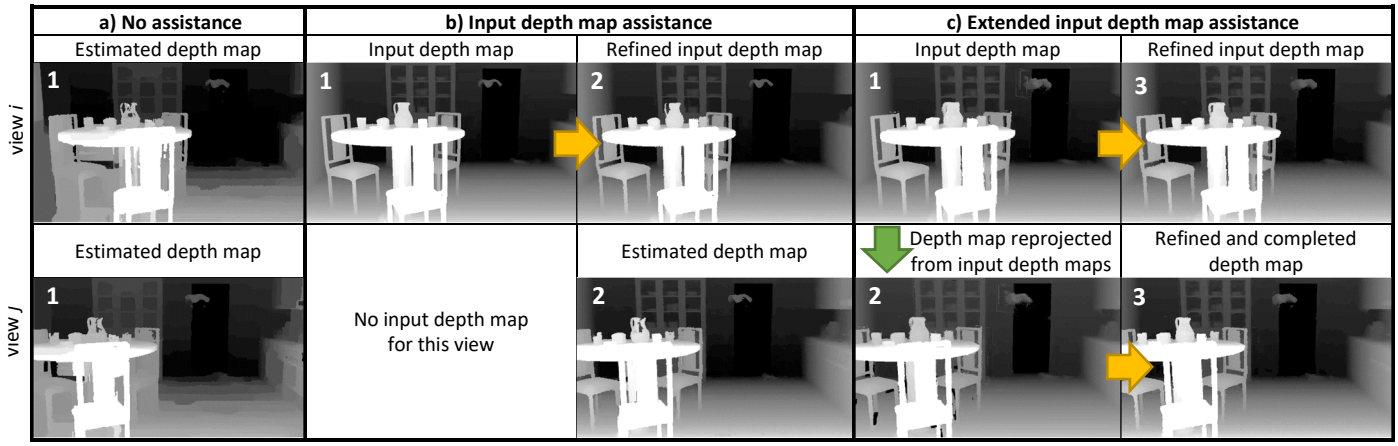


Fig. 4. Simplified overview of decoder-side depth estimation for two views, performed without assistance and for two proposed methods: a) depth maps for views *i* and *j* are estimated in the decoder (1); b) input depth map of view *i* decoded from the bitstream (1) is used in the joint process of estimation and refinement of depth maps for views *i* and *j* (2); c) decoded input depth map of view *i* (1) is reprojected to view *j* and becomes input depth map for view *j* (2), both input depth maps are used in the joint estimation and refinement process, resulting in depth maps for views *i* and *j* (3).

modified depth estimator, which refines them in the same way as input depth maps in the IDMA approach but also estimates depth for empty regions.

The proposals are based on the framework provided by the MIV compression standard [23]. Following subsections describe modifications proposed to the MIV encoder and decoder implemented in the Test Model for MPEG immersive video (TMIV) [30] and the required modifications of the depth estimator, which is treated as a part of the decoder.

B. Proposed modifications to the MIV encoder

1) Transmitted depth selection

The first step of the state-of-the-art MIV encoder is view labeling (Fig. 2). In this step, the encoder analyzes camera arrangement and derives a set of input views that carry the most non-redundant information (i.e., views with the smallest inter-view overlap). These views (called "basic views") are packed into atlases in their entirety, while others ("additional views") are either pruned and packed as a mosaic of smaller patches or completely discarded, depending on the MIV profile (Table I).

TABLE I
TYPES OF TRANSMITTED DATA FOR MIV MAIN AND MIV GA PROFILES.

View type	Input views	
	Basic	Additional
Transmitted data	MIV Main: texture, depth MIV GA: texture	MIV Main: texture, depth MIV GA: none

Unlike in typical approaches, where depth information is sent for all views (all basic and additional views) or not sent at all, in both proposed approaches (IDMA and eIDMA), only a subset of depth information is sent, resulting in texture atlases containing basic views and depth atlases, where the number of atlases for texture is higher than for depth. Fig. 5. shows an example of four atlases that consist of three texture and one depth atlases.

In the IDMA approach, the depth atlases contain full depth maps for views sent within the first *N* texture atlases among a total number of *M*, while depth information for the remaining basic views is skipped. In eIDMA, the depth atlases contain, besides the same information as in IDMA, a mosaic of smaller patches containing crucial and non-redundant depth fragments for basic views sent within the latter *M* – *N* texture atlases. Both approaches are designed to work with arbitrary *M* and *N* values, but the simplest case is when *N* equals one, and the below description is written based on this scenario.

Fig. 5: Example of four atlases produced by MIV encoder in the proposed methods: three texture atlases and one depth atlas: IDMA – only the yellow part, eIDMA – entire depth atlas (yellow and green parts).

In order to maximize profits gained by sending partial depth information, the MIV encoder has to properly select the depth maps which will be sent (additional yellow block in Fig. 6). In general, transmitted depth maps should cover the possibly largest area of the scene, and the overlap between them should be minimized. The MIV view labeling algorithm is already providing such a selection. Therefore, in both proposed IDMA solutions, the view labeler is run twice. In the first step, the basic views are selected from all input views. Next, basic views are analyzed in order to select "essential views" – views for which the depth maps are transmitted in full shape (Table II). The number of essential views may depend on the use case. In the simplest scenario, there are as many essential views as fit into an atlas.

Essential views are packed into the first texture atlas, while other basic views into the two remaining texture atlases (Fig.

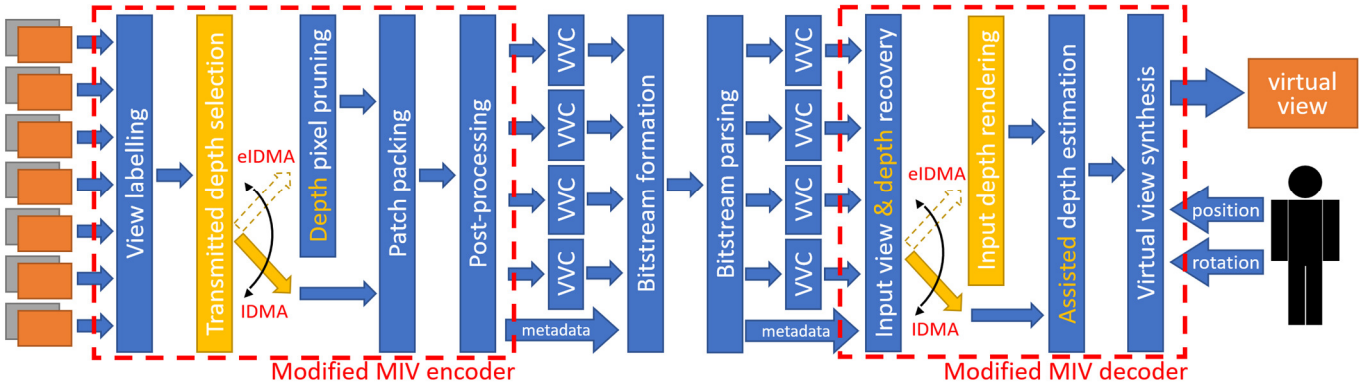


Fig. 6. The scheme of MIV encoder and decoder modified to handle two proposed methods. Modifications are indicated using yellow text and blocks.

7). In the IDMA approach, depth maps for views packed into the second and third atlas are estimated fully at the decoder side (Fig. 4b).

TABLE II
TYPES OF TRANSMITTED DATA FOR PROPOSED METHODS.

View type	Input views		
	Basic		Additional
	Essential	Non-essential	
Transmitted data	IDMA: texture, depth eIDMA: texture, depth	IDMA: texture eIDMA: texture, depth	None

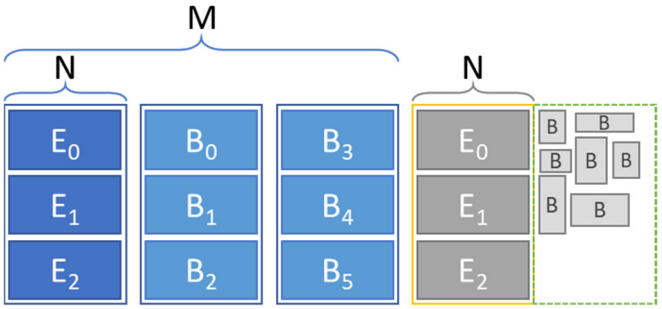


Fig. 7. Configuration of atlases in the proposed approach (blue – texture, grey – depth). M – total number of texture atlases (here – three), N – number of texture atlases for which the depth information is fully transmitted (one), E – essential views (three), B – basic non-essential views (six). The right part of the last atlas (with dotted boundary) is sent only in the eIDMA approach and contains partial depth information for all basic non-essential views.

During the experimental tests of our proposals, we identified that their efficiency is highly dependent on the characteristics of a multiview sequence being compressed. In the case of computer-generated sequences, where accurate depth maps are available, utilizing the eIDMA approach proves advantageous as it consistently delivers the highest quality. For natural content, where depth maps in the encoder are estimated based on captured views, the IDMA approach yields the best quality, as even if depth maps for these sequences are estimated using the most effective methods with optimized parameters, they cannot be considered as ground truth. Therefore, providing only partial assistance for such sequences is more preferable, as potential errors in the input depth maps will not be propagated to all views, which could occur in the case of eIDMA.

We provide an automatic selection mechanism for determining the most suitable method based on the content, dynamically choosing between the eIDMA and IDMA approaches. The required way for automatically indicating the quality of depth maps is already available in the MIV bitstream in the form of the automatically calculated depth quality flag, described in detail in [23]. Depth accuracy is used in MIV to determine how the encoder behaves and is also signaled to the decoder. A simple assessment of the geometry is applied based 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 flag indicates the low quality of depth maps, then the IDMA method is chosen (as it is the best for natural content). If the quality is assessed as high, then the eIDMA method is used.

2) *Depth pixel pruning*

The eIDMA approach does not need to estimate depth for all the pixels of these views, and a significant part of their depth maps is created by reprojection of depth from essential views (Fig. 4c). However, in such an approach, the transmission of depth maps only for essential views is insufficient. In this case, some areas of the depth map could not be properly estimated.

An example of this problem is presented in Fig. 8, where a depth map (Fig. 8b) was rendered using only information contained in the first atlas (Fig. 8a). Pixels of this view can be divided into three sets:

1. Significant part restored properly (areas visible in views contained in the first atlas – Fig. 8a),
2. An area with no information (white area: a large part of the floor, disocclusions behind chess pieces at the back),
3. An area with wrong information (e.g., the part with the head of the knight contains background information).

Depth values for the first set were reprojected correctly and match the reference depth map presented in Fig. 8c. For pixels from the second set, there is no depth information, so the depth will be estimated at the decoder side. However, pixels from the third set already have some reprojected information, and while this information is obviously wrong, the decoder cannot determine which fragments of the reprojected depth map belong to this set.

The decision whether a pixel of the basic (non-essential)

view belongs to the first, second, or third set is taken at the encoder side by reprojection of pixels from essential views. The pixel belongs to the first set if its depth value is equal or smaller (closer to the camera) than the co-located depth of reprojected pixels. If no information is reprojected to the position of the pixel (from any essential view), it belongs to the second set. Otherwise, the pixel belongs to the third set.

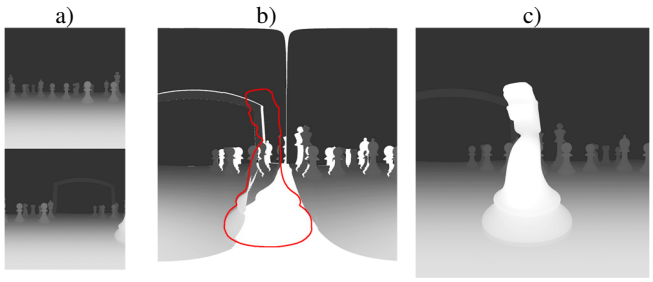


Fig. 8. Motivation for sending additional depth information. a) first depth atlas, b) depth for view rendered using information from the first atlas (silhouette of the knight is highlighted), c) reference depth for the view from the second atlas (not transmitted).

Therefore, as presented in Fig. 6, the eIDMA approach requires one additional step of depth pixel pruning, performed on depth maps for non-essential views. In this step, the modified MIV pruning algorithm is used: a pixel is pruned (removed) if it is inter-view redundant, as in MIV, but a second pruning condition is added, i.e., a pixel is pruned also if its depth can be estimated at the decoder side (on the basis of decoded textures). The only pixels which are preserved after the pruning step are the ones, which contain inestimable foreground objects, thus, the pixels for which the background from other views would be reprojected instead (e.g., the top part of the knight in Fig. 8b). These not-pruned pixels are packed into the bottom part of the depth atlas as a mosaic of patches (bottom of the fourth atlas in Fig. 5). The depth map rendered using additional depth patches is presented in Fig. 9b.

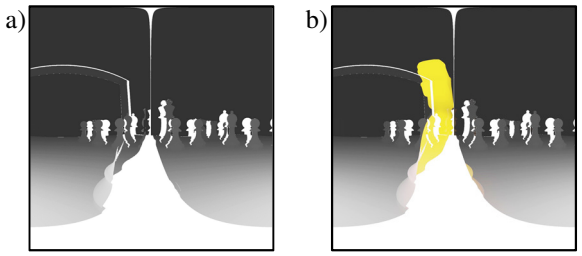


Fig. 9. a) A depth map rendered using depth maps from the first atlas and b) using depth maps from the first atlas together with additional depth patches; depth transmitted in additional patches was highlighted in yellow.

It should be noted that the height of the depth atlas in eIDMA is two times greater than for the IDMA approach. However, the resolution of depth atlases in MIV is by default reduced (twice in both directions) [23], [30], making the overall size of the depth atlas still much smaller than for each texture atlas.

C. Proposed modifications to the MIV decoder

1) Input view and depth recovery

In the first step, the modified MIV decoder (Fig. 6) unpacks received atlases and recovers all views and depth maps included in the bitstream. This step is performed using the state-of-the-art MIV algorithm [24] and results in the recovery of all basic views and all transmitted depth maps (depth for essential views in IDMA and depth for all basic and essential views in the eIDMA approach).

All recovered basic views are fed into the modified depth estimation algorithm (described in Section III.C.3). Recovered depth maps for essential views are processed depending on the chosen approach. In the IDMA approach, the depth estimator receives B recovered input views, and E recovered input depth maps, where B and E are the numbers of basic and essential views, accordingly, and $B > E$.

2) Input depth rendering

In the eIDMA approach, the depth estimation algorithm also uses B recovered input views, but the number of input depth maps is extended from E to B. Input depth maps for essential views are fed into the depth estimator in an unchanged form. For non-essential views, the recovered depth maps are mostly empty (Fig. 10a), as they initially contained redundant information, which was pruned at the encoder side. However, these areas can be rendered by reprojection of depth from input depth maps of other views, resulting in more complete depth maps (Fig. 10b), which are fed into the depth estimation algorithm in this form.

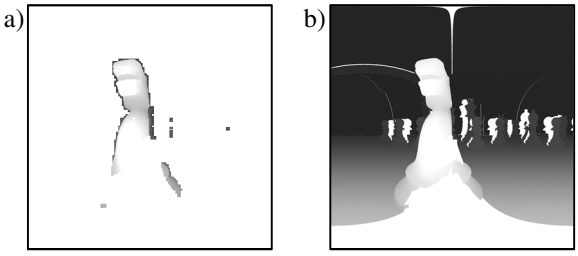


Fig. 10. A depth map for non-essential view; a) recovered, b) rendered by reprojection from other views and extended using the recovered depth map.

3) Assisted depth estimation

Besides changes in the MIV decoder, described in previous subsections, in order to implement the proposed compression methods, it was required to adapt some depth estimation method to be able to handle input depth maps. For this purpose, we decided to modify the Immersive Video Depth Estimation (IVDE [10]), which is the ISO/IEC MPEG Video Coding reference software for depth acquisition. This software has already been aligned with the MIV standard and meets the requirements imposed by the MIV on the depth estimation process [6], making it a natural choice for the scheme presented in this article.

Depth estimation in IVDE is based on cost function minimization, as defined in [10]. A minimum is determined using the graph cut algorithm [31]. Unlike other methods of depth map estimation based on graphs, in which graph vertices represent each pixel of the input views, each vertex corresponds to one superpixel [32]. To obtain cross-view consistency of the estimated depth maps, the cost of matching potentially corresponding points in adjacent views is not calculated

independently for each view but has been replaced by the cross-view matching cost defined between a pair of segments corresponding to the currently considered depth. Depth estimation is started with all superpixels assigned to the farthest depth level. The graph cut algorithm assigns points to the closer depth level in each subsequent iteration. Therefore, the estimator performs as many graph cut optimizations as there are levels of depth.

IVDE includes the method of temporal consistency enhancement in which some parts of depth maps can be copied from the previous frame and marked as unchangeable (by excluding the corresponding vertex from the optimization), while for the remaining parts, the depth estimation is performed as usual. This mechanism of excluding some vertices from the graph was utilized to handle input depth maps available in the decoder in order to use values from non-empty parts of input depth maps for the currently estimated depth map. If some area in the input map is empty (white area in Fig. 10b), then the depth for this area is estimated from scratch, and all depth levels are checked for these areas.

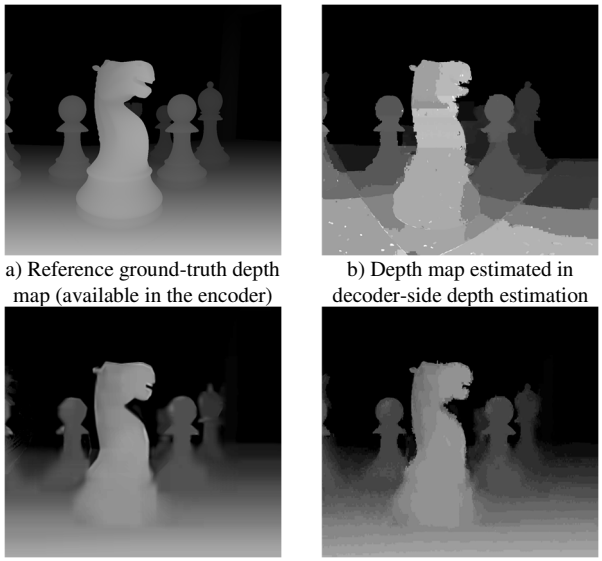
What is crucial in the depth estimation method implemented in IVDE, is that the process of depth optimization is global, i.e., depth maps for all views are estimated simultaneously in one common process. Therefore, if a set of high-quality depth maps is available and used as input depth maps, then the quality of depth maps for other views is also improved (Fig. 4). Moreover, the complexity of the estimation process is reduced, as the area for which the depth has to be estimated from scratch is significantly decreased.

The process described above provides utilization of input depth maps in the decoder-side depth estimation, however, it assumes that the quality of these depth maps will be sufficiently high. As discussed in Section I, the typical video codecs used for depth maps compression, also used in the MIV coding scheme, can lead to noticeable artifacts in the final, synthesized virtual views. Therefore, in order to reduce the compression-induced depth errors, the IVDE was modified to refine depth maps during the estimation process. This opens the possibility of using higher quantization when compressing depth maps – if less bits will be used for depth transmission, then the bitrate of textures will be increased, making the refinement easier to perform. The case of increasing quantization parameter for depth maps compression was also tested in experiments shown in Section IV.

In IVDE, the depth is estimated not for each pixel but for each segment (superpixel) calculated using texture information. In the beginning, for each pixel of a superpixel, it is checked what are the smallest and the largest depth values in the input depth map. These two values determine a small range of depth values for each superpixel that will be checked during the global optimization using graph cut. This way, even if the depth of some pixels is damaged during the compression, the depth values of adjacent pixels will be used to correct it.

Fig. 11 shows the fragment of the final estimated depth map when very high compression of depth maps was used. As can be observed, errors resulting from strong compression (blurred regions in the bottom) and depth reprojection are significantly reduced. When compared to the depth map estimated in the decoder-side depth estimation scheme (without input depth maps), the proposed eIDMA scheme still provides a depth map that is more similar to the depth map available in the decoder, even if high compression is introduced.

It should be noted that the depth map presented in Fig. 11c may seem visually more plausible than the one in Fig. 11d. However, the depth map before IVDE contains artifacts, which are extremely destructive in terms of the quality of rendered views presented to the final viewer – blurred edges. Such edges imply the appearance of disturbing artifacts (i.e., ghost edges [33]) in rendered views. As a result, in research on immersive video, the direct quantitative comparison of depth maps is rarely used as it does not express the quality of virtual views presented to the final user [34], as even in the case of using ground-truth depth maps, some artifacts in view synthesis can occur [35]. Therefore, in our study, the synthesis-based quality assessment is used in experimental comparison, with the methodology described in Section IV.B.



a) Reference ground-truth depth map (available in the encoder)
b) Depth map estimated in decoder-side depth estimation
c) Highly compressed input depth map in eIDMA (before IVDE)
d) Final depth map in eIDMA (after IVDE)

Fig. 11. Fragment of depth map: a) reference ground-truth; b) acquired in decoder-side depth estimation; c) compressed input depth map in eIDMA; d) depth map acquired by refinement of input depth map performed in decoder-side depth estimation process.

It should be noted that the proposed scheme is not restricted to be used with the described depth estimation method. While the efficiency of using the proposal with different estimators can vary, IDMA and eIDMA can be used together with any depth estimation method, similarly to the encoder-derived geometry features scheme (already tested with different estimators, e.g., IVDE [6] and DERS [13]).

D. Summary

The presented review of state-of-the-art compression schemes for immersive video shows the main disadvantages of

related work. In this section, we shortly summarize how our proposal deals with issues identified in Sections I and II:

- When the required bitrate is unavailable, depth maps can exhibit visual artifacts, blurring, and discontinuities around object boundaries. However, our approach includes depth refinement during the depth estimation process, eliminating the need for an additional computationally expensive refinement step.
- DSDE relies on high-quality depth estimation that is usually time-consuming, but our scheme significantly speeds up the estimation process as depth has to be estimated for smaller area.
- Unlike basic DSDE methods, we utilize information from high-quality depth maps typically available on the encoder side.
- The basic DSDE scheme encodes only full views in atlases, decreasing the amount of data that can be packed into atlases. In eIDMA, we use pixel pruning to allow packing of small non-redundant depth patches.
- Our approach uses standard video compression for depth maps, which is not possible when depth features are used to improve DSDE. Depth maps make it easier to represent complex structures, eliminating the need for bitrate wastage on signaling block division.
- Our scheme is based on the MIV scheme of coding by video pre-processing. Unlike previous top-performing encoders like MH-HEVC, our scheme allows for continuous improvements even after the completion of the MIV standard works.

IV. Experimental Results

A. Overview

In order to evaluate the performance of proposed compression methods, we conducted a comparison of IDMA and eIDMA with two state-of-the-art compression schemes for immersive video: the basic MIV Decoder-Side Depth Estimation anchor [36] and the MIV DSDE with Geometry Assistance SEI (encoder-derived features assistance) [28]. Section IV.B describes the methodology of performed experiments, while Section IV.C presents their results.

B. Methodology

The conditions of experiments are based on the common test conditions defined by ISO/IEC MPEG Video Coding to provide a fair comparison between methods for immersive video coding [37]. During the test, we utilize TMIV – Test Model for MPEG immersive video 11.0 [30], which implements the MPEG immersive video coding standard. In MIV DSDE and MIV DSDE with Geometry Assistance SEI, we use publicly available unmodified encoder and decoder, while for the IDMA and eIDMA, we utilize the proposal described in Section III. The following pixel rate constraints are imposed on all configurations:

- 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 [38]).
- 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.

For video compression, the VVenC codec [17], a fast implementation of VVC, is used. We use four rate points (RP) in all experiments: 3, 12, 22, and 38 Mbit/s, referred as RP4, RP3, RP2, and RP1, respectively. These values represent a practical range of bitrates used for the compression of immersive video, easily broadcasted using 5G networks [39]. The quantization parameter (QP) values used in encoding textures and depth atlases were tuned for all tested methods independently to match rate points as close as possible. All presented bitrates include summarized bitrates used for texture, depth, and other required metadata.

For depth estimation performed at the decoder side, we use Immersive Video Depth Estimation (IVDE [6], [10]), which is used in MIV experiments conducted by ISO/IEC MPEG Video Coding, as defined in common test conditions for MIV [37]. The same depth estimation method is used in all tested configurations, which ensures testing the coding scheme, not the depth estimation method itself. We use publicly available IVDE 7.0 [40], as it already provides support for encoder-derived features and input depth maps, implemented earlier by the authors of this paper.

After depth maps are estimated, the TMIV renderer is used to synthesize virtual views in the same position as all views of the used test sequences. In order to measure the quality of virtual views, the IV-PSNR metric [35] is used. The IV-PSNR values presented in the results are averaged for all views and test sequences. A brief summary of test sequences is available in Table III. In all experiments, 17 frames are used for the evaluation.

Configurations of tested compression schemes were as follows:

- MIV DSDE:
  - as in common test conditions for MIV [37],
- MIV DSDE with Geometry Assistance SEI:
  - as in the state-of-the-art approach [28] adapted in MIV [41], features calculated for a block of the maximum size of 64, quantization step for features equal to 256,
- IDMA and eIDMA: two texture-depth QP value schemes:
  - standard: depth $\mathrm{QP} = \max(1, [-14.2 + 0.8\, tex\ QP])$,
  - modified: depth $\mathrm{QP} = \max(1, 0.8\, tex\ QP)$.

Modified depth maps QP values are introduced to test if highly compressed depth can be refined in the depth estimation, allowing more bitrate for textures. Standard depth QP values are calculated as in common test conditions [37].

Tested methods were also evaluated in terms of their computational complexity. These results are provided in the form of runtime ratio when compared to no assistance DSDE. Experiments were conducted on a set of PCs with 3rd generation

AMD Ryzen Threadrippers equipped with 128 GB of RAM.

TABLE III
LIST OF TEST SEQUENCES.

Sequence	Source	Type		Resolution	Views
ClassroomVideo	[42]	ERP	CG	4096 × 2048	15
Chess	[43]	ERP	CG	2048 × 2048	10
Hijack	[44]	ERP	CG	4096 × 2048	10
Museum	[44]	ERP	CG	2048 × 2048	24
Group	[45]	Perspective, convergent	CG	1920 × 1080	21
Fencing	[46]	Perspective, convergent	NC	1920 × 1080	10
Fan	[47]	Perspective, planar	CG	1920 × 1080	15
Kitchen	[48]	Perspective, planar	CG	1920 × 1080	25
Mirror	[49]	Perspective, planar	NC	1920 × 1080	15
Carpark	[50]	Perspective, planar	NC	1920 × 1088	9
Frog	[51]	Perspective, planar	NC	1920 × 1080	13
Hall	[50]	Perspective, planar	NC	1920 × 1088	9
Street	[50]	Perspective, planar	NC	1920 × 1088	9
Painter	[52]	Perspective, planar	NC	2048 × 1088	16

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

C. Results

The results of performed experiments are shown in Fig. 12 in the form of plots showing the average IV-PSNR of rendered views obtained using tested compression methods for different rate points (bitrates averaged for all tested sequences). For low bitrate, the results for basic DSDE (without assistance) confirmed its high quality, presented earlier in other works [5], [6]. Nevertheless, for high bitrates, the final quality of DSDE is one of the worst of the tested methods.

The use of encoder-derived features [28], the first method to utilize information from depth maps available in the encoder to improve DSDE, increased the quality for medium and high bitrates when compared to basic DSDE. On the other hand, for low bitrates, the size of metadata required to send features (more than 20% of the whole bitstream – Table IV) negatively influences the quality of encoded textures, decreasing the final quality of rendered virtual views.

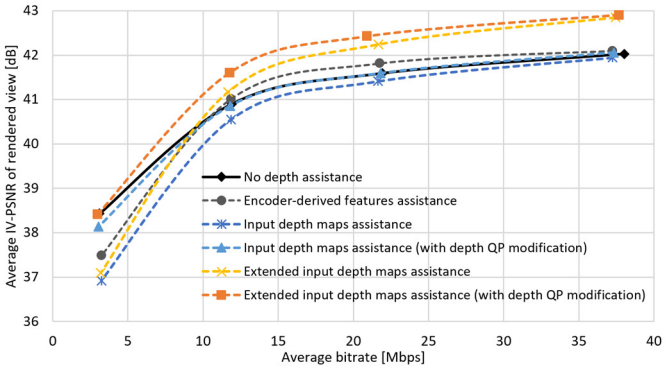


Fig. 12. Average IV-PSNR of rendered views achieved by tested methods; shown bitrates include all texture, depth, and metadata.

Input depth map assistance was shown to be the worst, however, when the modified depth QP is used, the quality increases for all bitrates. For medium and high bitrates, the average quality is comparable to basic DSDE, but, as presented in Table V, the runtime reduction for the decoder is significant. A slight increase in runtime for the encoder is also observed, but the complexity of atlas encoding in DSDE is negligible when compared to video encoding.

TABLE IV
DISTRIBUTION OF BITRATE USED FOR TEXTURE, DEPTH, AND METADATA.

No assistance [6]

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	38.008	0.000	0.009	38.017	99.98%	0.00%	0.02%
RP2	21.915	0.000	0.009	21.924	99.96%	0.00%	0.04%
RP3	11.923	0.000	0.009	11.931	99.93%	0.00%	0.07%
RP4	3.105	0.000	0.009	3.114	99.72%	0.00%	0.28%

Encoder-derived features assistance [28]

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	36.653	0.000	0.689	37.342	98.15%	0.00%	1.85%
RP2	21.075	0.000	0.689	21.764	96.83%	0.00%	3.17%
RP3	11.112	0.000	0.689	11.801	94.16%	0.00%	5.84%
RP4	2.521	0.000	0.689	3.210	78.54%	0.00%	21.46%

Input depth maps assistance

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	32.500	4.735	0.008	37.243	87.26%	12.71%	0.02%
RP2	17.667	3.977	0.008	21.653	81.59%	18.37%	0.04%
RP3	9.095	2.749	0.008	11.853	76.74%	23.19%	0.07%
RP4	1.993	1.236	0.008	3.237	61.56%	38.18%	0.26%

Input depth maps assistance with depth QP modification

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	35.686	1.635	0.008	37.329	95.60%	4.38%	0.02%
RP2	20.487	1.326	0.008	21.821	93.89%	6.08%	0.04%
RP3	10.777	1.017	0.008	11.803	91.31%	8.62%	0.07%
RP4	2.655	0.369	0.008	3.032	87.56%	12.16%	0.28%

Extended input depth maps assistance

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	30.543	6.877	0.018	37.437	81.58%	18.37%	0.05%
RP2	16.370	5.277	0.018	21.665	75.56%	24.36%	0.08%
RP3	8.136	3.494	0.018	11.648	69.85%	30.00%	0.15%
RP4	1.668	1.514	0.018	3.199	52.13%	47.31%	0.55%

Extended input depth maps assistance with depth QP modification

Rate point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	35.399	2.258	0.018	37.675	93.96%	5.99%	0.05%
RP2	19.160	1.731	0.018	20.909	91.64%	8.28%	0.08%
RP3	10.494	1.263	0.018	11.775	89.12%	10.73%	0.15%
RP4	2.496	0.475	0.018	2.988	83.52%	15.89%	0.59%

It is important to note that the software used in this study, including the IVDE depth estimator, TMIV, and VVenC, were not optimized for low computational complexity but are the current implementations used for academic and standardization purposes. The real-time implementations of required processes can already be found, e.g., for MIV bitstream decoding and computationally expensive virtual view rendering [53], which in the presented experiments took about 20 seconds per view).

Extended assistance provides better quality than previously discussed methods for medium and high bitrates and has the fastest decoding and rendering time. When depth QP is modified, the method shows the best results among all tested configurations for all rate points, showing the high performance of depth refinement performed using modified depth estimator.

When analyzing the virtual view quality in Fig. 12 together with the content of the encoded bitstream in Table IV, it can be seen that for the best methods at each rate point (eIDMA with depth QP modification for all RPs and additionally eIDMA in RP1 and no depth assistance and IDMA with depth QP modification in RP4), the fraction of bitrate used for depth

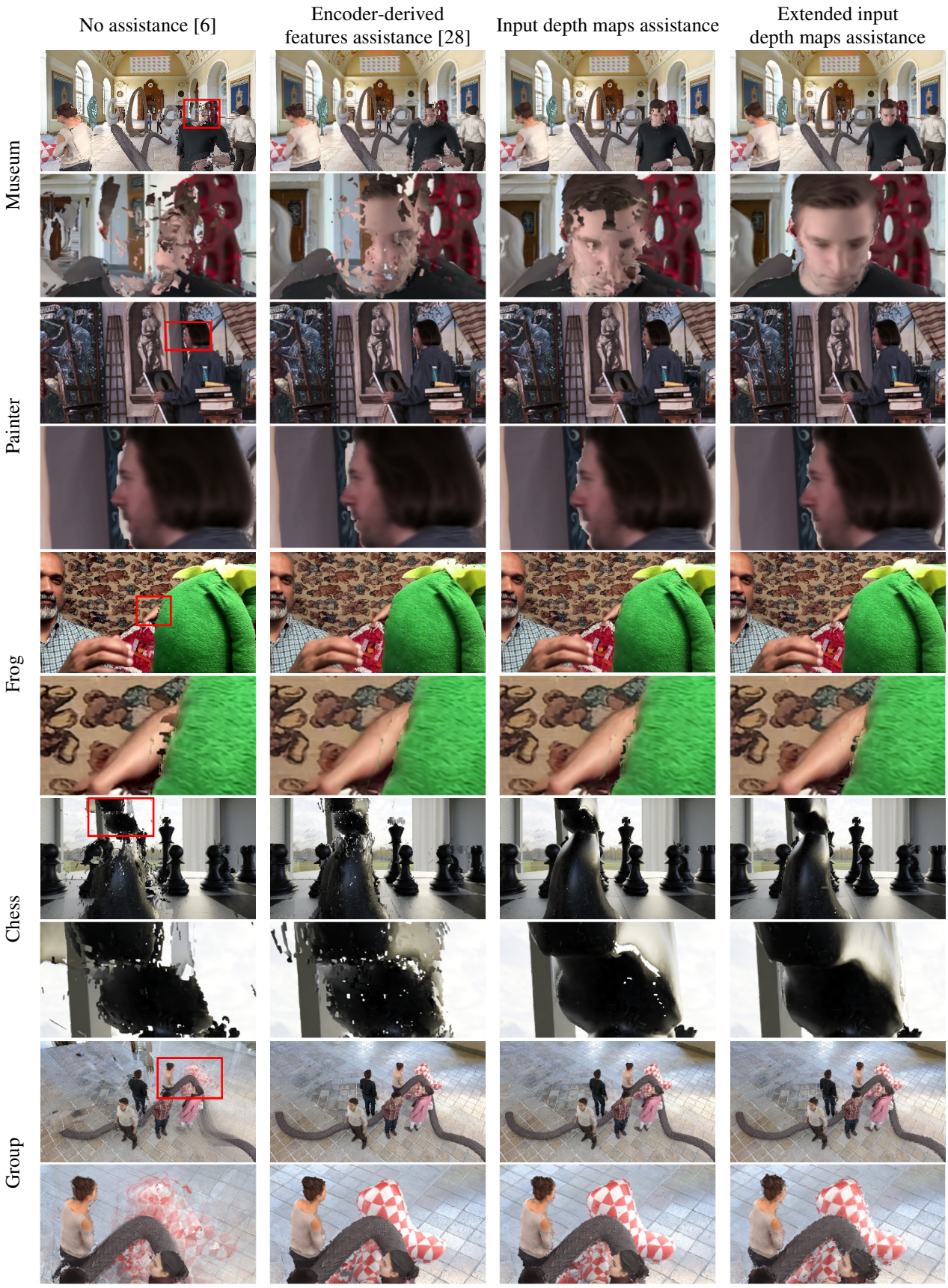


Fig. 13. The subjective comparison of tested methods for fragments of selected viewport synthesized between positions of input views.

information (in the form of encoded depth maps or features included in metadata) does not exceed 20%. It ensures a decent quality of textures, enabling estimation of better depth maps and, in the end, higher quality of synthesized views, as this step is influenced both by better textures and depth maps. When no depth information is being sent, the quality of textures is very high, however, possible significant errors in depth maps (resulting from occlusions, non-Lambertian surfaces) affect the final quality of virtual views and prevent the decoder from providing high quality, especially for high bitrates.

It can also be observed in the provided visual comparison of virtual views obtained for tested methods, presented in Fig. 13. While the quality for most of the areas in virtual views is similar for different methods, the enlarged fragments show fragments of views for which the erroneously-estimated depth maps significantly decreased the observed quality in basic DSDE. Providing encoder-derived features [28] or the proposed input depth map assistance considerably improves the reconstruction quality in these areas.

As presented in Fig. 13, in most of the cases, proposed IDMA and eIDMA approaches also outperform the encoder-derived features. The only test sequence for which the proposed method performed worse than [28] was Group. This computer-generated sequence has a relatively low number of objects, and most fragments in depth maps are flat. Therefore, ranges of possible depth levels are very similar for neighboring blocks, making it easy to compress these depth map features in this coding scheme, increasing the final quality of rendered views. However, the subjective quality gain over the proposal is negligible when compared to the poor quality of virtual views rendered using the basic DSDE approach. Moreover, the decoding process in both IDMA and eIDMA methods is significantly faster than using the encoder-derived features.

TABLE V
THE RUNTIME OF TESTED METHODS; SHOWN PERCENTAGES PROVIDE RUNTIME RATIO COMPARED TO DSDE WITHOUT ASSISTANCE.

Type of assistance	Average with standard deviation of runtime per one frame [s]			
	Encoding			Decoding & rendering of a view
	Atlas encoding	Video encoding	Overall	
No assistance [6]	1.5±0.3	228.1±104.9	229.6±104.8	152.3±62.2
Features [28]	1.9±0.6 (126%)	228.1±104.9 (100%)	231.9±105.1 (101%)	83.8±33.7 (55%)
IDMA	7.7±6.9 (518%)	216.7±101.4 (95%)	227.3±99.3 (99%)	73.1±25.5 (48%)
IDMA + QP mod.	7.7±6.9 (518%)	235.0±106.1 (103%)	245.7±103.0 (107%)	74.6±27.4 (49%)
eIDMA	66.0±60.2 (4481%)	191.6±68.1 (84%)	261.7±106.1 (114%)	48.7±12.8 (32%)
eIDMA + QP mod.	66.0±60.2 (4481%)	212.1±100.6 (93%)	277.8±120.2 (121%)	48.7±13.4 (32%)

When the objective quality results are shown independently for CG and natural content (Fig. 14 and Fig. 15, respectively), it can be seen that the quality of depth maps available in the encoder has a considerable influence on the differences between efficiencies of tested methods. In the case of CG sequences, for which the ground-truth depth maps are available, it is beneficial to use the eIDMA approach, as it provides the best quality, independently of used depth QP values. However, for natural content, for which depth maps available in the encoder are estimated basing on the acquired views, the best quality is observed for the IDMA approach. In this case, even if estimated using the best methods with fine-tuned parameters, depth maps cannot be considered a ground-truth. Therefore, providing only partial assistance for such sequences is more favorable, as possible errors in input depth maps will not be reprojected to all views, which could be the case in eIDMA.

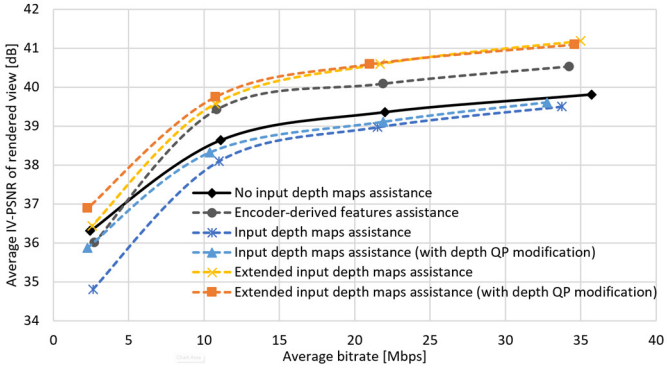


Fig. 14. Results for CG sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

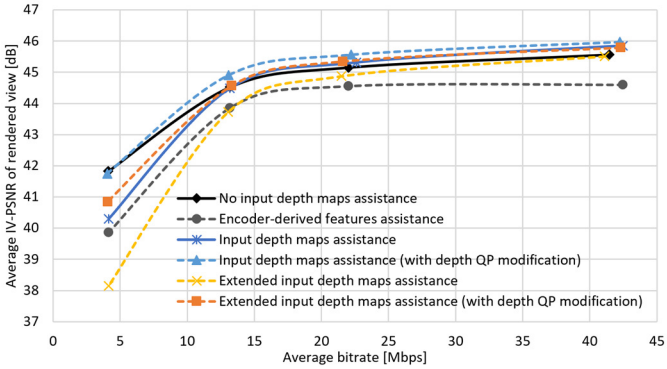


Fig. 15. Results for natural sequences: average IV-PSNR of rendered views achieved by tested methods; bitrates include all texture, depth, and metadata.

Although extended assistance with modified depth QP seems to be the best compromise, the encoder was improved to automatically select the best method, depending on the currently encoded content (adaptive IDMA scheme, described in Section III.B.1).

Fig. 16 shows the performance of such a mixed method. As it can be seen, such a proposal of adaptive IDMA performs much better than other tested methods, merging two solutions into one robust solution for DSDE-based immersive video coding.

The final results are shown in Table VI. This table shows BD-rate (Bjøntegaard delta [54]), i.e., the percentage change in the bitrate required to achieve the same quality for tested coding techniques in comparison with the reference method (no assistance DSDE). The results show a significant bitrate saving of over 50% for the final adaptive IDMA method. Combining this result with the faster decoding of the proposed methods shows the proposal's advantages over state-of-the-art DSDE.

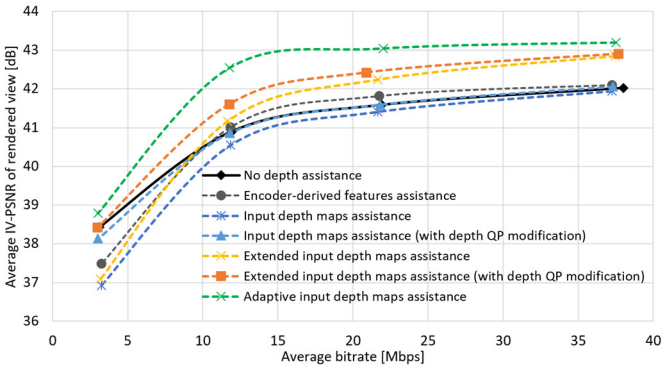


Fig. 16. Average IV-PSNR of rendered views achieved by tested methods and proposed adaptive input depth maps assistance for four tested bitrates; bitrates include all texture, depth, and metadata.

TABLE VI
BD-RATE OF TESTED METHODS VERSUS NO ASSISTANCE DSDE [6].

Type of assistance	BD-rate versus no assistance [6]
Encoder-derived features [28]	0.70%
IDMA	32.70%
IDMA + depth QP mod.	3.50%
eIDMA	-2.70%
eIDMA + depth QP mod.	-27.40%
Adaptive IDMA	-53.00%

V. CONCLUSIONS

The paper describes a novel method of increasing the efficiency of the DSDE approach in immersive video transmission. When compared to the state-of-the-art DSDE techniques, where the textures of views and camera parameters are transmitted to the decoder, which estimates the 3D scene representation from scratch, we have proposed to additionally send a subset of depth maps available at the encoder side. In the decoder, this additional geometry information assists the depth estimation process, increasing the overall efficiency.

The proposed approach significantly speeds up the decoding process when compared to the state-of-the-art DSDE approach [6], allowing for estimating depth maps even three times faster. Moreover, the proposed input depth map assistance achieves a better quality of final virtual views, as the depth estimator is guided by the original geometry transmitted to the decoder.

We proposed two ways of transmitting the input depth information within an MPEG immersive video (MIV) bitstream, meeting different requirements of the practical, immersive video systems. The advantage of the first proposal, where the depth information is sent for only a subset of transmitted input views, is a relatively fast decoding process and quality improvement over the state-of-the-art DSDE approach, visible especially for encoding of natural content. In the second, extended approach, additional depth patches for remaining views are being transmitted, exploiting pruning and packing algorithms of the MPEG immersive video encoder. Such an approach allows for an even faster decoding process and further improvement of the quality of viewports presented to the final user of the immersive video system.

Previous works have proven that the DSDE approach achieves satisfactory quality in compressed immersive video, especially for low-bitrate systems. However, the depth map estimation process is time-consuming, and developing a real-time immersive video encoder working in the DSDE mode is challenging. Therefore, the authors believe that the proposed input depth map assistance approach is a step in the direction of the development of practical immersive video systems. Moreover, using the proposed approach, the decoder is able to reproduce a good-quality representation of the 3D scene even for challenging content, including non-Lambertian surfaces, numerous disocclusions, and areas with inestimable depth (e.g., for areas visible only in one transmitted view).

The proposal, besides changing the main principle of DSDE, that the depth maps are not available in the decoder, changes also the assumptions of immersive video coding, that there cannot be any depth in the transmitted bitstream that does not correspond to texture. Our scheme proposes more flexible encoding, e.g., by opening the possibility of using depth acquired from sensors such as Kinect [55], which are very popular among consumers [56]. This acquired depth can be easily used as an input depth map and improve the quality of depth for other views. Functionalities of the proposal enable many further improvements, e.g., much faster estimation of depth only for missing areas in reprojected depth maps, performed using deep-learning-based methods or by simple inpainting.

MPEG Video Coding experts appreciated the flexibility, usefulness, and novelty of proposed IDMA-based approaches, therefore, these encoding methods will comply with a new DSDE profile [57], [58] of the incoming second edition of the MIV standard [59], as agreed during the 140th MPEG meeting in October 2022.

REFERENCES

[1] J.-B. Jeong, et al., "Towards 3DoF+ 360 Video Streaming System for Immersive Media," *IEEE Access*, vol. 7, pp. 136399-136408, 2019.
[2] D. Mi et al., "Demonstrating Immersive Media Delivery on 5G Broadcast and Multicast Testing Networks," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 555-570, 2020.
[3] M. Tanimoto et al., "FTV for 3-D Spatial Communication," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, April 2012.
[4] A. Vetro et al., "3D-TV Content Storage and Transmission," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 384-394, 2011.
[5] V.K.M. Vadakital et al., "The MPEG Immersive Video Standard—Current Status and Future Outlook," *IEEE MultiMedia*, vol. 29(3), 2022.
[6] 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, 2022.
[7] Y.-L. Chan et al. "Overview of current development in depth map coding of 3D video and its future," *IET Signal Proc.*, vol. 14(1), pp. 1-14, 2020.
[8] M. Wien et al., "Standardization Status of Immersive Video Coding," *IEEE J. Emer. and Sel. Topics in Circ. and Syst.*, vol. 9(1), pp. 5-17, 2019.
[9] J. Samelak et al., "Advanced HEVC Screen Content Coding for MPEG Immersive Video," *Electronics*, vol. 11, no. 23, pp. 4040, 2022.
[10] D. Mieloch et al., "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," *IEEE Access*, vol. 8, pp. 5760-5776, 2020.
[11] D. Mieloch, A. Dziembowski, M. Domański, "Depth Map Refinement for Immersive Video," *IEEE Access*, vol. 9, pp. 10778-10788, 2021.
[12] M. Ibrahim et al., "Depth map artefacts reduction: a review," *IET Image Processing*, vol. 14, no. 12, pp. 2630-2644, 2020.
[13] P. Garus et al., "Bypassing Depth Maps Transmission for Immersive Video Coding," in *PCS 2019*, Ningbo, China, 2019, pp. 1-5.
[14] A. Dziembowski et al., "The influence of a lossy compression on the quality of estimated depth maps," in *IWWSIP 2016 Conf.*, 2016, pp. 1-4.

[15] P. Garus et al., "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.
[16] G. J. Sullivan et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Tr. Circ. & Syst. for V. Tech.*, vol. 22(12), 2012.
[17] A. Wieckowski et al., "VVenC: An Open and Optimized VVC Encoder Implementation," in *ICMEW 2021 Conf.*, 2021, pp. 1-2.
[18] G. Sullivan and T. Wiegand, "Video compression – from concepts to the H.264/AVC standard," Proceedings of the IEEE, vol. 93, 2005.
[19] M. Domański et al., "Universal Modeling of Monoscopic and Multiview Video Codecs with Applications to Encoder Control," in *2021 IEEE International Conference on Image Processing*, pp. 2144-2148, 2021.
[20] 2021 TV Video Specifications. Accessed: Jan. 11, 2021. [Online]. Available:https://developer.samsung.com/smarttv/develop/specifications/media-specifications/2021-tv-video-specifications.html .
[21] H.264/H.265 Video Codec Unit. Accessed: Jan. 11, 2021. [Online]. Available: xilinx.com/products/intellectual-property/v-vcu.html
[22] 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.
[23] J. Boyce, et al., "MPEG Immersive Video coding standard," *Proceedings of the IEEE*, vol. 109, no. 9, p. 1521-1536, 2021.
[24] "Text of ISO/IEC FDIS 23090-12 MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/N0111, Online, 2021.
[25] M. Milovanović et al., "Patch Decoder-Side Depth Estimation in Mpeg Immersive Video," in *ICASSP 2021*, pp. 1945-1949, 2021.
[26] S.L. Ravi et al., "A Study of Conventional and Learning-Based Depth Estimators for Immersive Video Transmission," in *IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, Shanghai, China, 2022.
[27] B. Szydełko, et al., "Recursive block splitting in feature-driven decoder-side depth estimation", *ETRI Journal*, vol. 44, pp. 38– 50, 2022.
[28] G. Clare et al., "Combination of m56626 and m56335 for Geometry Assistance SEI message," ISO/IEC JTC1/SC29/WG4/M56950, 2021.
[29] P. Garus et al., "Motion Compensation-based Low-Complexity Decoder Side Depth Estimation for MPEG Immersive Video," in *IEEE Int. Workshop on Multimedia Signal Proc. (MMSP)*, Shanghai, China, 2022.
[30] "Test Model 11 for MPEG Immersive Video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0142, Online, October 2021.
[31] Y. Boykov et al., "Fast approximate energy minimization via graph cuts," *IEEE Tr. on Pattern An. and Mach. Int.*, vol. 23(11), pp. 1222-1239, 2001.
[32] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple noniterative clustering," in *CVPR 2017,* Honolulu, HI, USA, Jul. 2017.
[33] A. Dziembowski et al., "Multiview synthesis – improved view synthesis for virtual navigation," in *32nd Picture Coding Symposium (PCS)*, Nürnberg, Germany, 2016.
[34] F. Shao et al., "Depth Map Coding for View Synthesis Based on Distortion Analyses," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 106-117, March 2014.
[35] 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, 2022.
[36] "Report of MPEG immersive video CTC anchor generation," ISO/IEC JTC1/SC29/WG4 MPEG2022/N0234, Online, July 2022.
[37] "Common Test Conditions for MPEG Immersive video," ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0143, Online, October 2021.
[38] ISO/IEC 23008-2, Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding.
[39] C. Colman-Meixner et al., "Deploying a Novel 5G-Enabled Architecture on City Infrastructure for Ultra-High Definition and Immersive Media Production and Broadcasting," *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 392-403, 2019.
[40] "MPEG-I Visual / IVDE – Gitlab", available online, accessed 28.09.2022: https://gitlab.com/mpeg-i-visual/ivde/-/tree/v7.0.
[41] "Text of ISO/IEC DIS 23090-5 Visual Volumetric Video-based Coding and Video-based Point Cloud Compression 2nd Edition," ISO/IEC JTC1/SC29/WG7 MPEG2021/N00188, Online, July 2021.
[42] B. Kroon, "3DoF+ Test Sequence ClassroomVideo," ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415, San Diego, April 2018.
[43] L. Ilola et al., "New Test Content for Immersive Video – Nokia Chess," ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787, Geneva, Sep. 2020.
[44] R. Doré, "Technicolor 3DoF+ Test Materials," ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349, San Diego, March 2018.
[45] R. Doré, G. Briand, and F. Thudor, "InterdigitalGroup Content Proposal," ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731, Online, June 2020.
[46] 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.
[47] R. Doré, G. Briand, and F. Thudor, "InterdigitalFan Content Proposal for MIV," ISO/IEC JTC1/SC29/WG11 MPEG/M54732, Online, June 2020.
[48] 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.
[49] 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.
[50] D. Mieloch et al., "Natural Outdoor Test Sequences," ISO/IEC JTC1/SC29/WG11 MPEG2019/M51598, Brussels, January 2020.
[51] B. Salahieh et al. "Kermit Test Sequence for Windowed 6DoF Activities," ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748, Ljubljana, July 2018.
[52] D. Doyen et al., "Light Field Content from 16-camera Rig," ISO/IEC JTC1/SC29/WG11 MPEG2017/M40010, Geneva, January 2017.
[53] M. Chen et al., "Simplified carriage of MPEG immersive video in HEVC bitstream," Proc. SPIE, vol. 11842, Aug. 2021, Art. no. 118420C
[54] G. Bjoentegaard, "Calculation of average PSNR differences between RD-Curves," *ITU-T VCEG Meeting*, Austin, USA, 2001.
[55] X. Ye et al., "Computational Multiview Imaging with Kinect," *IEEE Transactions on Broadcasting*, vol. 60, no. 3, pp. 540-554, 2014
[56] A. Doumanoglou et al., "Quality of Experience for 3-D Immersive Media Streaming," *IEEE Tr. on Broadcasting*, vol. 64, no. 2, pp. 379-391, 2018.
[57] A. Dziembowski et al., "MIV Decoder-Side Depth Estimation profile," ISO/IEC JTC1/SC29/WG4 MPEG VC M60667, Mainz, 10.2022.
[58] D. Mieloch, A. Dziembowski, J. Y. Jeong and G. Lee, "On the future of decoder-side depth estimation in MPEG immersive video coding," in *2023 Data Compression Conference (DCC)*, Snowbird, UT, USA, 2023.
[59] "Preliminary WD4 of ISO/IEC 23090-12 MPEG immersive video Ed. 2," ISO/IEC JTC1/SC29/WG4 MPEG VC N0269, Mainz, 10.2022.

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 immersive media technologies. He has been involved in several projects focused on multiview video. His professional interests also include depth estimation, and camera calibration.

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.

Dominika Klóska was born in 1998. She received her B.Sc. and M.Sc. degrees from Poznań University of Technology in 2021 and 2022 respectively. Currently she works at the Institute of Multimedia Telecommunications where she is involved in ISO/IEC MPEG activities such as MPEG immersive video coding standard and the development of other immersive media technologies.

Błażej Szydełko received the master's degree in electronics and telecommunications from Poznan University of Technology (PUT), Poland in 2022. Currently, he is a research and teaching assistant at the Institute of Multimedia Telecommunications at PUT. His professional interests are related to immersive video technologies, especially multiple-camera system calibration and depth estimation.

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.