On the future of decoder-side depth estimation in MPEG immersive video coding

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

*Institute of Multimedia Telecommunications, Poznan University of Technology
Polanka 3, Poznań, 61-151, Poland
{dawid.mieloch, adrian.dziembowski}@put.poznan.pl

+Electronics and Telecommunications Research Institute
Daejeon, 34129, Republic of Korea
{jyj0120, gslee}@etri.re.kr

Abstract: This paper presents the new profile to supersede the existing Geometry Absent profile supported in the MPEG immersive video (MIV) coding standard. The proposed MIV Extended Decoder-Side Depth Estimation (MIV DSDE) profile was developed to cover more diverse use cases and applications based on the decoder-side depth estimation scheme and allow for further improvements of the efficiency of incoming MIV ed. 2, even after the standard will reach its final stage. This paper presents also the first review of state-of-the-art compression methods and predicted future schemes based on decoder-side depth estimation.

1. Introduction

One of the most commonly used data representations for supporting 6DoF (Degrees of Freedom) immersion [1] is multiview video which consists of multiple videos and their corresponding depth maps [2]. Since this format cannot support smooth view transition by itself, the decoder side has to be accompanied by smart view interpolation techniques such as DIBR (Depth Image-Based Rendering). However, massive data volume remains the biggest barrier to be properly served in different aspects of the existing multiview processing workflow, e.g., due to bandwidth limitations and specifications of the currently available hardware. Therefore, dedicated compression methods are necessary.

Several compression methods have been already developed for multiview video and the most representative ones are 3D-HEVC and MV-HEVC [3] which utilize inter-view redundancy among overlapping views. However, they are not widely adopted by immersive video systems due to limitations resulting from their design. For example, 3D-HEVC can properly compress the video only in a perspective format, leaving other common formats used for representing immersive video unsupported. In addition, this codec only works properly for linear camera arrangement, leading to strict restrictions on the freedom of the user's viewpoint rotation. In the case of MV-HEVC, even though depth maps can be available in the bitstream, this information is not utilized for removing the redundant data in views being sent.

By considering the above requirements, the development of the MPEG immersive video (MIV) [4] coding standard started in 2019, and it reached an official stage in 2022 [5]. MIV is an extension of the V3C standard [6], designed for encoding various types of volumetric content. Besides the MVD (handled by MIV), it also standardizes the bitstream format of point clouds, however, in this paper, we only consider MIV-related representations. In brief, MIV reduces inter-view redundancy based on the process called pruning, designed for removing the redundant pixels by cross-projection of views. Then, the preserved information is clustered into several patches (fragments of pruned views), and they are packed into a much smaller number of mosaic videos named atlases [4], [7] (Fig. 1). Since

atlases are still maintained as 2D shape, MIV is codec-agnostic, thus the atlases may be encoded using any video encoder, such as AVC, HEVC [8], or VVC [9].

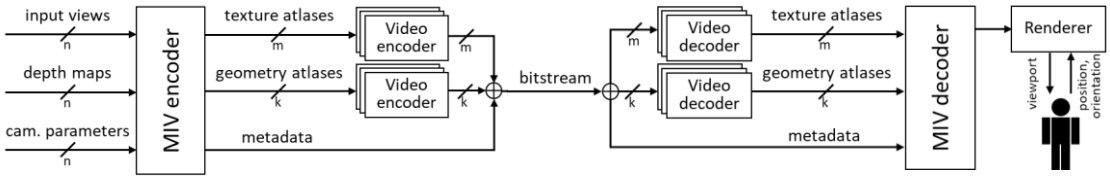

Figure 1: Simplified scheme of an MIV encoder and decoder.

MIV contains diverse profiles [5] that enable the encoding of different types of input videos (e.g., multi-planar images – MPI [10]) or its use in particular applications. One of the profiles, MIV Geometry Absent (MIV GA), assumes replacing the transmission of depth maps by their estimation from decoded views in the client or the cloud, implementing the so-called decoder-side depth estimation (DSDE) encoding scheme.

In this approach, the complexity of the whole compression process is largely shifted from the video-capturing side to the decoder side. While the concept is relatively old, as its first introduction can be traced to the development of first multiview codecs [11], it was used more often in the context of inter-view prediction, not for obtaining geometry required for virtual view synthesis. Nevertheless, the scheme was admitted to being too computationally expensive to be used by the techniques existing at that time [12]. Much later, DSDE was tested also during the early phases of the immersive video coding standard development [13], [14], and as part of an architecture of simple free-viewpoint television systems [15].

Growing computational power available in the recent decoding devices and following industrial interest in this scheme entailed the successful introduction of this scheme into the newest immersive video coding standard. Since then, on the wave of intensive works on MIV ed. 1, numerous new approaches were presented. New proposals tend to further reduce the computational complexity of the process and to provide a higher quality of reconstructed depth. Unfortunately, many of them cannot be implemented using existing profiles defined in the MIV standard.

First of all, this paper provides an overview of DSDE-related works presented to this day in the literature and by standardization groups. Moreover, in this contribution, we also describe a new "MIV Decoder-Side Depth Estimation" (MIV DSDE) profile, proposed by the Authors, which is designed to be a successor of the MIV Geometry Absent (MIV GA) profile for MIV ed. 2. When compared to MIV GA, where geometry sub-bitstreams are completely absent in the bitstream, proposed MIV DSDE profile is much more flexible and covers all already presented (and possible future) use cases and applications.

2. Existing and potential use cases of decoder-side depth estimation

This section presents the description and discussion on the already existing methods proposed for compressing immersive video based on the DSDE scheme. In addition, by considering the current development direction of MIV ed. 2 [16], we also try to foresee the encouraging potential use cases of the DSDE.

2.1. MIV Geometry Absent

The MIV Geometry Absent profile follows basic principles of DSDE, i.e., no geometry is present in the encoded bitstream, and it is estimated in the decoder from decoded multiple

views. This scheme was shown to be codec-agnostic [17], as MIV Main, so any modern video encoder can be used to compress texture atlases, (Fig. 2). The depth estimation is the non-normative part of the MIV decoder, therefore, any method which meets criteria described in [18] can be utilized. When comparing results from [18] and [19], it can be seen that the differences between the efficiency of depth map estimators, usually measured as the quality of synthesized virtual view using estimated depth maps and uncompressed input view, are similar when depth estimation is performed on decoded views recovered from atlases.

When compared to MIV Main, MIV GA provides around 30 % of bitrate reduction for perspective content [16] and was shown to be very efficient for low bitrates [18]. Naturally, it presents higher computational complexity at the decoder side, but when deep learning-based methods are applied for depth estimation, the decoding time can be reduced to be 7 times slower than MIV Main [18].

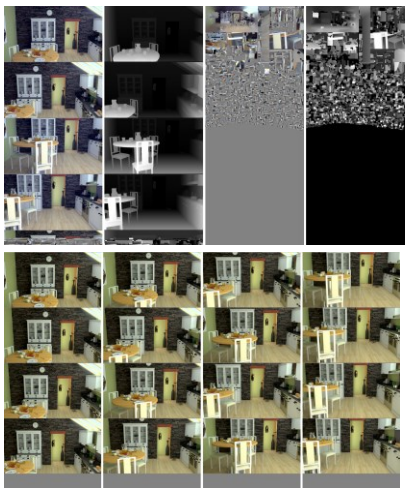

Figure 2: Example of atlases produced by encoder in MIV Main profile (top – atlases contain whole views, their fragments, and corresponding depth maps) and MIV GA profile (bottom – atlases contain whole views).

2.2. Geometry Assistance SEI

Even though the encoding process of MIV GA described in Section 2.1 is very simple, the computational burden for depth estimation is very high, and, importantly, high-quality depth maps from the encoder side cannot be used. To tackle this issue, the MIV standard supports Geometry Assistance SEI (GA SEI) [5], [20] which can be used to enable the transmission of a set of encoder-derived features from the depth maps to improve the quality of depth maps acquired in decoder-side depth estimation and to speed up this process [13]. It was also shown in [10] that this scheme can be efficiently used to construct multiplane images [16] at the decoder side.

By sending additional information on 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 applied for static regions, this SEI was shown to provide the objective quality similar as in DSDE with the time of decoding at least halved [18]. Depending on the methods, the depth estimation can be almost 20 times faster [13], nevertheless, in this scheme it is required to implement features usage functionalities in the depth estimator. Therefore, this fact highly narrows the set of suitable methods.

In general, features can use from 1 to 2 Mbit/s of available bandwidth [18], so for low bitrates (less than 5 Mbit/s), the bitrate left for textures can be very low, visibly hampering their quality. Furthermore, as features are only 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 increases the bitrate twice [21]).

The described drawbacks were addressed in the new version of this assistance, i.e., the Extended Geometry Assistance SEI (EGA SEI) adopted to be a part of MIV ed. 2 [22].

Besides typical geometry assistance (features sent for all views, no recursion [23]), the new SEI includes the recursive splitting of blocks [21], the possibility of sending encoder-derived features for a subset of views [24], and the merge of previously listed schemes [25]. Different levels of details for different views (quantization step, block width, number of splits set per view) are also allowed. The new SEI also provides the possibility of adding new schemes of feature extraction (not only the block-based rectangular grid), making it more future-proof than its predecessor.

2.2.1 Motion Compensation-based Decoder Side Depth Estimation

The method presented in [26] utilizes the depth estimation performed both on the encoder and the decoder side. The encoder-side estimation utilizes motion vectors extracted from encoded texture videos to decide if for given block the depth map should be estimated from decoded views, as in default DSDE, or recovered from previous depth maps using motion compensation. It is shown that this proposal moves a significant part of complexity from decoder to the encoder, as for the optimal parameters it provides around 20 times faster decoding. Although the proposal is described as proof of concept and does not specify the details of how the described block-based decision is signaled in the bitstream, we argue that the concept is very similar to the structure of GA SEI and could be implemented using the available skip flag [18] to determine for which blocks depth has to be estimated.

2.3. MIV Main without depth transmission

This scheme was presented in [27]. The method is based on recovering pruned views, but no temporal redundancy removal [28] is used, as it makes patches to be not fully occupied. Depth maps are not included in a bitstream, therefore, occupancy of patches, usually embedded in depth maps, cannot be extracted. The pruned views recovered by the decoder were fed into an unmodified depth estimator (Immersive Video Depth Estimation – IVDE, the reference depth estimation method of ISO/IEC MPEG Video Coding [18]). The experimental results showed around 10 % bitrate reduction when compared to the MIV Main [27]. We predict that this scheme can be enhanced by using the external occupancy video (not available for MIV Main, but available for MIV Extended). In this case, temporal redundancy removal [28], can be still used, possibly reducing the required bitrate.

2.4. MIV Main with depth refinement

We argue that a depth refinement is a form of depth estimation, therefore, such post-processing will be treated in further considerations as decoder-side depth estimation. The most straightforward approach would be to include the depth refinement in the MIV Main configuration without pruning. Besides packing views into atlases, this scheme can be treated as equivalent to simulcast coding, as only whole views and their depth maps are included in the bitstream, and no inter-view dependencies are utilized.

Other foreseen scheme would work similarly as in MIV Main without depth transmission (described in Section 2.3), so the pruning can be also utilized, but it would be necessary to recover, besides pruned input views, also pruned depth maps, which can be later refined.

Multiple depth refinement methods are designed to be useful in the reduction of compression-related errors [29]. The immersive-video-related method from [30] can be also utilized, as this refinement method does not utilize any color information. As the quality of encoded views in DSDE can be very low, especially for very low bitrates, the

use of this refinement, independent of the quality of input views, can be particularly desirable in such applications.

2.5. Input depth map assistance

Based on the scheme of encoding a part of depth maps in order to improve the depth estimation performed at the decoder side [31]. 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. It is possible by utilizing the modified depth estimation method based on global multi-view optimization. Results provided in [31] show that with this assistance, the DSDE is around 30 % faster than MIV GA, on average.

2.5.1 Extended input depth map assistance

In addition, the speed and quality aspect of the above-described method can be further enhanced by reprojecting the set of available input depth maps to other views for which the input depth maps are not available for all pixels [32]. These reprojected depth maps are also fed into the modified depth estimator which refines them in the same way as input depth maps in the previous approach, but also estimates new depth only for empty regions. As presented in [32], this scheme is on average 60 % faster than MIV GA and provides much better quality on demanding content (e.g., omnidirectional) [32]. Fig. 3 shows a set of fragments of virtual views showing subjective quality of decoded views for extended input depth map assistance and other previously described schemes of compression.

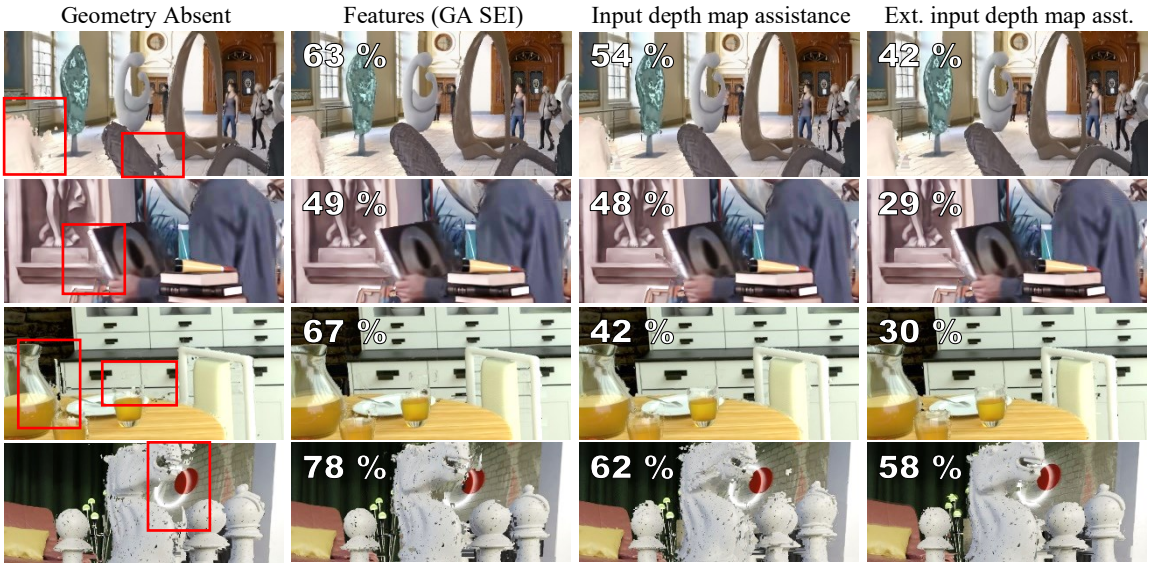

Fig. 3: Fragments of virtual views shared with [25], [31], [32] and the decoding time presented as a percentage of decoding time in the MIV Geometry Absent approach.

2.5.2 Faster extended input depth map assistance

We assume a possible close-to-real-time DSDE scenario with the reprojection of the input depth maps, as described above, but without the reestimation of the transmitted and reprojected depth. This scheme would require only filling empty regions of reprojected depth maps, which can be done by depth estimator, or just by simple depth-based inpainting. Nevertheless, the quality of the resulting depth maps could be worse than in the previous scenario, as no refinement is applied to the transmitted data.

2.6. Usage of depth sensors

It is assumed in the requirements for the new MIV ed. 2 that support for depth sensors will be provided [16]. Now, in MIV Main, depth already can be sent without attributes, but this data is not used for rendering. We foresee that depth from sensors could be reprojected to other views and used as an input depth map, similarly as in the input depth map assistance scheme.

2.7. Comparison of various approaches

To list the functionalities required by described methods, we present them in Table 1. As described in Section 3, most of the presented approaches are not compatible with the available MIV ed. 1 profiles, showing the need for a new profile enabling the support of these use cases.

Table 1: Functionalities required by presented methods.

Functionality \ Section	MIV Geometry Absent (and GA SEI)	MIV GA with geometry assistance SEI	MIV Main w/o depth transmission (embedded occupancy)	MIV Main w/o depth transmission (with external occupancy)	MIV Main with depth refinement	Input depth map assistance	Extended input depth map assistance	Faster extended input depth map assistance	Usage of depth sensors	MIV Main (no DSDE)
	2.1	**2.2**	**2.3**		**2.4**	**2.5**	**2.5.1**	**2.5.2**	**2.6**	**-**
Decoder-side depth estimation	yes	yes	yes	yes	partial [1]	yes	yes	partial [2]	yes	no
Texture video transmission	yes	yes	yes	yes	yes	yes	yes	yes	partial [3]	yes
Texture pruning in encoder	no	no	yes	yes	yes	no	no	no	no	yes
Geometry (depth) video transmission	no	no	no	no	yes	partial [4]	partial [4]	partial [4]	partial [4]	yes
Geometry pruning in encoder	no	no	no	no	yes	no	yes	yes	no	yes
External occupancy video transmission	no	no	no	yes	no	no	no	no	no	no
Additional (non-video) depth information	no	yes	no	no	no	no	no	no	no	no
Depth reprojection in decoder	no	no	no	no	no	no	yes	yes	yes	no

[1] Only depth refinement (reestimation); [2] Depth estimated only for fragments of the depth maps; [3] Bitstream contains geometry videos which do not correspond to any texture video; [4] Bitstream contains texture videos which do not correspond to any geometry video.

3. Proposed MIV Extended Decoder-Side Depth Estimation sub-profile

This section contains a technical description of the MIV Extended Decoder-Side Depth Estimation (MIV DSDE) sub-profile, including syntax adopted to the incoming MIV ed. 2 standard [22] and its interpretation, as well as three additional proposed flags, which allow for covering all use cases presented in Section 2.

In general, MIV DSDE extends the capabilities of MIV GA, by allowing for transmitting additional data, i.e., geometry and occupancy video for particular atlases (cf. *vps_geometry_video_present_flag* and *vps_occupancy_video_present_flag* in Table 2). Besides the possibility of using the external occupancy video, MIV DSDE also allows for embedding occupancy within the geometry video (cf. *vme_embedded_occupancy_enabled_flag* in Table 2). In MIV DSDE, it is also possible to transmit geometry video which does not correspond to any texture video (i.e., when *ai_attribute_count* is equal to 0, Table 2), what is required for transmitting depth acquired by depth sensors.

Formally, the MIV DSDE is a sub-profile of the MIV Extended profile, which is the most flexible and comprehensive among available profiles. However, MIV DSDE restricts several syntax elements to be adjusted to the decoder-side depth estimation applications.

Table 2: Allowable values of syntax elements for different MIV profiles (an extract from Table A-1 in [5] with proposed modifications highlighted).

	Profile name			
Syntax element	MIV Main	MIV Extended	**MIV DSDE**	MIV GA
ptl_profile_toolset_idc	64	65		66
vps_miv_extension_present_flag	1	1	**1**	1
ai_attribute_count[*atlasID*]	0, 1	0, 1, 2	**0, 1**	1
vps_occupancy_video_present_flag[*atlasID*]	0	0, 1	**0, 1**	0
vps_geometry_video_present_flag[*atlasID*]	1	0, 1	**0, 1**	0
vme_embedded_occupancy_enabled_flag	1	0, 1	**0, 1**	0
casme_decoder_side_depth_estimation_flag	**0**	**0, 1**	**1**	**1**

Since the MIV DSDE is dedicated for MIV, not the entire V3C, it requires to use the MIV extension of the V3C parameter set [6], signaled by setting the *vps_miv_extension_present_flag* to 1.

Together with the new MIV DSDE profile, we have proposed to add the following five flags into the existing MIV syntax: *casme_decoder_side_depth_estimation_flag*, *mvp_depth_reprojection_flag*, *mvp_reestimate_all_geometry_flag*, *mvp_keep_transmitted_geometry_flag*, *mvp_keep_reprojected_geometry_flag*. The first two proposed flags are already adopted into MIV ed. 2 [22]. Other three flags may be included in the MIV syntax in the future, either in the *miv_view_params_list*, as proposed, or as an SEI message.

The first proposed flag, *casme_decoder_side_depth_estimation_flag* is defined within the MIV extension of the common atlas sequence parameter set (Table 3). It signals whether the depth should be estimated or reestimated at the decoder side. As presented in Table 2, *casme_decoder_side_depth_estimation_flag* is set to 0 for MIV Main and may be set to 0 for non-DSDE scenarios of the MIV Extended profile, decreasing the number of unnecessary DSDE-related flags in the bitstream. For two profiles based on the DSDE approach: MIV Geometry Absent and MIV DSDE, the flag is set to 1.

Table 3: Syntax of the MIV extension of the common atlas sequence parameter set (CASPS) – table from Section 8.3.2.5 of [5] with proposed additional flag (highlighted).

casps_miv_extension( )	**Descriptor**
casme_depth_low_quality_flag	u(1)
casme_depth_quantization_params_present_flag	u(1)
casme_vui_params_present_flag	u(1)
casme_decoder_side_depth_estimation_flag	u(1)
if(casme_vui_params_present_flag)	
vui_parameters( )	

In addition to *casme_decoder_side_depth_estimation_flag*, four additional DSDE-related flags are defined within the MIV view params list (Table 4). The *mvp_depth_reprojection_flag* is already adopted to MIV ed. 2, while others may be added in the future to cover faster depth estimation (cf. Section 2.5.2).

Table 4: Syntax of the MIV view params list – updated table from Section 8.3.2.6.2 of [5]: adopted to MIV ed. 2 (light grey) and potential future improvements (dark grey).

miv_view_params_list( )	**Descriptor**
mvp_num_views_minus1	u(16)
...	
if(casme_decoder_side_depth_estimation_flag)	
mvp_depth_reprojection_flag	u(1)
mvp_reestimate_all_geometry_flag	u(1)
if(mvp_reestimate_all_geometry_flag == 0)	
for(v = 0; v <= mvp_num_views_minus1; v++)	
mvp_keep_transmitted_geometry_flag[v]	u(1)
mvp_keep_reprojected_geometry_flag[v]	u(1)

The *mvp_depth_reprojection_flag* equal to 1 indicates that the decoder creates geometry data for all views by reprojection of transmitted geometry values before the depth estimation process, making it possible to use additional depth sensors (cf. Section 2.6) or the extended input depth map assistance approach (cf. Section 2.5.1).

The *mvp_reestimate_all_geometry_flag* equal to 1 indicates that all geometry values will be reestimated in the decoder (even if they were transmitted within the geometry video sub-bitstream). Setting the flag to 1 allows for reducing the negative impact of video coding artifacts, as decoded depth maps can be used as input in depth estimation or refinement process. If the flag is set to 0, some geometry values can be kept as they are, on the basis of two further flags, allowing for significantly faster decoding. These two flags are *mvp_keep_transmitted_geometry_flag*[*v*] and *mvp_keep_reprojected_geometry_flag*[*v*], and are set independently for each view *v*. If the first one is set to 1, the depth estimator does not modify depth values transmitted within geometry video sub-bitstreams. If the second one is set to 1, also the depth values reprojected from other views are kept, and the depth estimator estimates only the holes within depth maps. Such an approach may be used for faster depth map estimation, allowing for achieving real-time decoder-side depth estimation in the reasonable future (cf. Section 2.6).

Configuration of the MIV syntax elements for different DSDE applications, presented in Section 2 is presented in Table 5. For example, for the extended input depth map assistance approach, there are four atlases. Atlas 0 contains both texture and geometry information (*ai_attribute_count*[0] = 1 and *vps_geometry_video_present_flag*[0] = 1), atlases 1 and 2 have no geometry, and atlas 3 has geometry only (*ai_attribute_count*[3] = 0 and *vps_geometry_video_present_flag*[3] = 1). In this approach, there is no external occupancy video (*vps_occupancy_video_present_flag*[*atlasId*] = 0 for all atlases), as the occupancy is embedded within the geometry video (*vme_embedded_occupancy_enabled_flag*). As indicated by *mvp_depth_reprojection_flag* = 1, all transmitted geometry values are reprojected between views before the depth estimation.

Table 5: Examples of MIV syntax elements configuration for existing and potential compression methods of MPEG immersive video standard.

	MIV Geometry Absent	MIV GA with geometry assistance SEI	MIV Main w/o depth transmission (embedded occupancy)	MIV Main w/o depth transmission (with external occupancy)	MIV Main with depth refinement	Input depth map assistance	Extended input depth map assistance	Faster extended input depth map assistance	Usage of depth sensors	MIV Main (no DSDE)
Section / **Syntax element**	**2.1**	**2.2**	**2.3**		**2.4**	**2.5**	**2.5.1**	**2.5.2**	**2.6**	**-**
ai_attribute_count[*atlasID*]	1,1,1,1 [1]		1,1	1,1	1,1	1,1,1	1,1,1,0	1,1,1,0	1,1,1,0	1,1
vps_geometry_video_present_flag[*atlasID*]	0,0,0,0		0,0	0,0	1,1	1,0,0	1,0,0,1	1,0,0,1	0,0,0,1	1,1
vps_occupancy_video_present_flag[*atlasID*]	0,0,0,0		0,0	1,1	0,0	0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0
vme_embedded_occupancy_enabled_flag	0		0	0	1	1	1	1	1	1
casme_decoder_side_depth_estimation_flag	1		1	1	1	1	1	1	1	0
V3C SEI payloadType	-	133/134 [2]	-	-	-	-	-	-	-	-
mvp_depth_reprojection_flag	0		0	0	0	0	1	1	1	-
mvp_reestimate_all_geometry_flag	1		1	1	1	1	1	0	0	-
mvp_keep_transmitted_geometry_flag[*v*]	-		-	-	-	-	-	1	1	-
mvp_keep_reprojected_geometry_flag[*v*]	-		-	-	-	-	-	0/1 [3]	0/1	-

[1] Notation "1,1,1,0" means, that a flag is true for atlases 0, 1, and 2, and false for atlas 3.
[2] payloadType equal to 133 indicates geometry assistance SEI [23], 134: extended geometry assistance SEI [20].
[3] The flag may be set to 0 or 1 for each view *v*, depending on its quality and desired computational time reduction.

4. Conclusions

This paper presents the first review of state-of-the-art compression methods based on decoder-side depth estimation. The comparison is done in the context of the MPEG immersive video standard to show its current status and define possible directions of the development to be performed for the new edition of this standard. We also predict some future use cases and solutions, which were not described in the literature and were not considered within standardization groups to show the flexibility of presented schemes and to encourage other researchers to provide further improvements in this part of immersive video compression.

We proposed a new MIV Extended Decoder-Side Depth Estimation profile which allows for using all reviewed compression methods but also is as future-proof as possible. Besides providing the framework for future DSDE-related methods, the proposal, as all existing MIV profiles, is codec-agnostic, so any video compression still can be used for the compression of produced atlases. The proposal also does not refer to any particular depth estimator (or depth refinement method), therefore, it is still a non-normative part of the compression workflow. All these features enable the continuous improvement of the efficiency of incoming MIV ed. 2, even after the standard will reach its final stage.

Both the already adopted parts of the profile and the additional flags were proposed by the Authors during the 140th MPEG meeting [33].

5. Acknowledgment

The work was supported by the Ministry of Education and Science of Republic of Poland.

6. References

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

[2] 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, 2011.

[3] G. Tech et al., "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," *IEEE Tr. on Circ. and Syst. for Vid. Tech.*, vol. 26(1), pp. 35-49, 2016.

[4] J. Boyce et al., "MPEG Immersive Video coding standard," *Proceedings of the IEEE*, vol. 109, no. 9, p. 1521-1536, 03.2021.

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

[6] ISO/IEC 23090-5, Information technology — Coded Representation of Immersive Media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC).

[7] "Test Model 11 for MPEG Immersive video," *Doc. ISO/IEC JTC 1/SC 29/WG 04 N 0142*, October 2021, Online.

[8] G. Sullivan et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Tr. on Circuits and Systems for Video Tech.*, vol. 22(12), pp. 1649-1668, 2012.

[9] B. Bross et al. "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Tr. on Circ. and Syst. for Vid. Tech.*, vol. 31, no. 10, Oct. 2021.

[10] P. Garus et al., "Decoder Side Multiplane Images using Geometry Assistance SEI for MPEG Immersive Video," in *MMSP 2022*, Shanghai, China, 2022.

[11] Y.S. Ho and K.J. Oh, "Overview of Multi-view Video Coding," in *2007 14th International Workshop on Systems, Signals and Image Processing*, pp. 5-12, 2007.
[12] A. Vetro, "Summary of BoG Discussion on View Interpolation Prediction," *ISO/IEC JTC1/SC29/WG11, Doc. JVT-W133*, San Jose, USA, 2007.
[13] P. Garus et al., "Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?," *IEEE T. Circ. & Syst. for Vid. Tech.*, vol. 32(5), 05.2022.
[14] L. Jorissen et al., "[FTV AhG] Soccer Light Field Interpolation Applied on Compressed Data," *Doc. ISO/IEC JTC1/SC29/WG11, MPEG2016/M37674*, San Diego, Feb. 2016.
[15] A. Dziembowski et al., "The influence of a lossy compression on the quality of estimated depth maps," in *IWSSIP Conference*, 2016, pp. 1-4
[16] V.K.M. Vadakital et al., "The MPEG Immersive Video Standard—Current Status and Future Outlook," *IEEE MultiMedia*, vol. 29(3), pp. 101-111, 2022.
[17] A. Grzelka et al., "The Study of the Video Encoder Efficiency in Decoder-side Depth Estimation Applications," *WSCG 2022*, Plzen, Czech Republic, May 2022.
[18] D. Mieloch et al., "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," *IEEE T. Circ. and Syst. for Vid. Tech.*, vol. 32(9), 09.2022.
[19] S.L. Ravi et al., "A Study of Conventional and Learning-Based DepthEstimators for Immersive Video Transmission," in *MMSP 2022*, Shanghai, China, 2022.
[20] A. Dziembowski et al., "[MIV] Extended geometry assistance SEI," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M60248*, Online, 18-22.07.2022.
[21] B. Szydełko et al., "Recursive block splitting in feature-driven decoder-side depth estimation," *ETRI Journal*, vol. 44, pp. 38– 50, 2022.
[22] "Preliminary WD4 of ISO/IEC 23090-12 MPEG immersive video Ed. 2," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC N0269*, Mainz, Germany, October 2022.
[23] G. Clare, et al. "[MIV] Combination of m56626 and m56335 for Geometry Assistance SEI message," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M56950*, Online, Apr. 2021.
[24] B. Szydełko, et al. "Partial geometry assistance information", *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M58047*, Online, October 2021,
[25] B. Szydełko, et al. "Effectiveness of recursive splitting in feature extraction for subset of views," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M58334*, Online, October 2021.
[26] P. Garus et al., "Motion Compensation-based Low-Complexity Decoder Side Depth Estimation for MPEG Immersive Video," in *MMSP 2022*, Shanghai, China, 2022.
[27] M. Milovanović, F. Henry, M. Cagnazzo, and J. Jung, "Patch Decoder-Side Depth Estimation in MPEG Immersive Video," in *ICASSP 2021,* pp. 1945-1949, 2021.
[28] A. Dziembowski et al. "Spatiotemporal redundancy removal in immersive video coding," *Journal of WSCG*, vol. 30, no. 1-2, pp. 54-62, 2022.
[29] M. Ibrahim, Q. Liu, R. Khan, J. Yang, E. Adeli, and Y. Yang, "Depth map artefacts reduction: a review," *IET Image Processing*, vol. 14, no. 12, pp. 2630-2644, 2020.
[30] D. Mieloch, A. Dziembowski and M. Domański, "Depth map refinement for immersive video," *IEEE Access*, vol. 9, pp. 10778-10788, 2021.
[31] D. Klóska et al., "Decoder-side depth estimation with input depth assistance," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M58048*, Online, 11-15.10.2021.
[32] D. Mieloch et al., "[MIV] Decoder-side depth estimation with extended input depth assistance," *Doc. ISO/IEC JTC1/SC29/WG4 MPEG VC M59516*, Online, Apr. 2022.
[33] A. Dziembowski, D. Mieloch, J.Y. Jeong, G. Lee, "MIV Decoder-Side Depth Estimation profile," *ISO/IEC JTC1/SC29/WG4 MPEG VC M60667*, Mainz, 10.2022.