Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video

Dawid Mieloch, Patrick Garus, Marta Milovanović, Joël Jung, *Member, IEEE*, Jun Young Jeong, Smitha Lingadahalli Ravi, and Basel Salahieh, *Senior Member, IEEE*

***Abstract*— This paper presents the overview and rationale behind the Decoder-Side Depth Estimation (DSDE) mode of the MPEG Immersive Video (MIV) standard, using the Geometry Absent profile, for efficient compression of immersive multiview video. A MIV bitstream generated by an encoder operating in the DSDE mode does not include depth maps. It only contains the information required to reconstruct them in the client or in the cloud: decoded views and metadata. The paper explains the technical details and techniques supported by this novel MIV DSDE mode. The description additionally includes the specification on Geometry Assistance Supplemental Enhancement Information which helps to reduce the complexity of depth estimation, when performed in the cloud or at the decoder side. The depth estimation in MIV is a non-normative part of the decoding process, therefore, any method can be used to compute the depth maps. This paper lists a set of requirements for depth estimation, induced by the specific characteristics of the DSDE. The depth estimation reference software, continuously and collaboratively developed with MIV to meet these requirements, is presented in this paper. Several original experimental results are presented. The efficiency of the DSDE is compared to two MIV profiles. The combined non-transmission of depth maps and efficient coding of textures enabled by the DSDE leads to efficient compression and rendering quality improvement compared to the usual encoder-side depth estimation. Moreover, results of the first evaluation of state-of-the-art multiview depth estimators in the DSDE context, including machine learning techniques, are presented.**

***Index Terms*— Depth map, immersive video, video codecs, video processing, cloud computing.**

Manuscript received 15 December 2021; revised 25 February 2022; accepted 17 March 2022. Date of publication 28 March 2022; date of current version 6 September 2022. The work of Dawid Mieloch was supported by the Ministry of Education and Science, Poland. This article was recommended by Associate Editor G. Pastuszak. *(Corresponding author: Dawid Mieloch.)*

Dawid Mieloch is with the Institute of Multimedia Telecommunications, Poznań University of Technology, 60-965 Poznań, Poland (e-mail: dawid.mieloch@put.poznan.pl).

Patrick Garus and Smitha Lingadahalli Ravi are with the Orange Labs, 35512 Cesson-Sévigné, France (e-mail: patrick1.garus@orange.com; smitha.lingadahalliravi@orange.com).

Marta Milovanović is with the Orange Labs, 35512 Cesson-Sévigné, France, and also with the LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: marta.milovanovic@orange.com).

Joël Jung is with the Tencent Media Laboratory, Palo Alto, CA 94306 USA (e-mail: joeljung@tencent.com).

Jun Young Jeong is with the Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea (e-mail: jyj0120@etri.re.kr).

Basel Salahieh is with Vimmerse Inc., San Jose, CA 95110 USA (e-mail: basel.salahieh@vimmerse.net).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2022.3162916.

Digital Object Identifier 10.1109/TCSVT.2022.3162916

I. Introduction

THE ability to change the point of view in three-dimensional scenes is the primary functionality offered by so-called immersive video services. While the parallax effect allows the viewer to fully experience the virtual reality with either head-mounted or light-field displays, legacy two-dimensional displays can still be utilized to show a desired viewing position and orientation (viewport) chosen with any interactive input device.

As immersive video applications gain more and more interest in the video processing community, an increase in standardization efforts is also noticeable [76], as the transmission of immersive content requires a very efficient representation. Even if immersive video makes usage of diversified systems for the acquisition and the presentation, the commonalities shared by these multimedia systems made it possible to create a versatile method for their compression.

The MPEG Immersive Video (MIV) standard [1] is the outcome of a collective industry effort to support immersive media access and delivery, a critical milestone for the emerging immersive ecosystem. The first edition of the MIV standard [12], released at the MPEG 135th meeting in July 2021, is based on the Visual Volumetric Video-based coding (V3C) standard [56] which defines the commonalities between MIV and the MPEG Video-based Point Cloud Coding (V-PCC). The MIV standard only defines the bitstream format and the decoding processes besides the supported profiles. The associated reference software, known as the Test Model for MPEG Immersive Video (TMIV) [10], covers, in addition, the non-normative parts which include the encoding and rendering processes.

It is well understood that independent compression of multiple views and depths [77] results in high bitrates. Moreover, a new critical constraint in immersive video services is the so-called pixel rate: it reflects the number of pixels to be decoded per second in order to initiate the rendering of a given virtual view. It is usually determined by practical resources of hardware decoders. For instance, the HEVC Main 10 profile at level 5.2 [59] specifies a limitation that is compatible with nowadays decoders (*e.g.* [57], [58]): assuming a 4096×2048 video at 30 fps, the number of decoder instantiations cannot exceed four. When applied to immersive video, this means that the bitstream cannot include more than two views and their corresponding depth maps.

Moreover, while modern universal video encoders provide good encoding performance for typical video content [48], depth map encoding is more efficient when dedicated solutions are used [49], making it inappropriate to use the same encoder for both purposes.

MV-HEVC and 3D-HEVC [82], which may be inappropriately seen as predecessors of MIV, did not address the described limitations, therefore, the development of such extensions of Versatile Video Coding (VVC) is not currently intended. Even though 3D-HEVC encoder implementations are still actively improved by some researchers (*e.g.,* by faster machine-learning-based depth map encoding [84], or introduction of joint source-channel coding [83]), it is still not suitable for modern immersive video applications, as 3D-HEVC is not compatible with non-linear arrangements of cameras, vertically displaced views, and omnidirectional content, significantly reducing its usability. To comprehend the extent of these limitations, it can be noted that in the experiments performed in Section IV, 10 out of 15 test sequences cannot be used with this codec. In conclusion, MIV cannot indeed be considered as the technical successor of 3D-HEVC, because of its fundamentally different premises and orientation on practical feasibility.

A recently researched approach further addresses the above-mentioned downsides of previous encoding techniques and consists in replacing the transmission of depth maps by its estimation from decoded views in the client or in the cloud [15]. Such an approach allows to include twice as many views into the bitstream as in typical compression of views and depth with MIV, with the same pixel rate. This makes it much easier for the renderer to provide a satisfactory quality of the synthesized view because the number of input views highly influences both the fidelity of the view synthesis [60], [61], and the depth estimation [3]. Without transmitting the depth, the complexity is shifted from the capture side to the decoder side. This approach, referred to as decoder-side depth estimation (DSDE), was introduced during the early phases of the immersive video coding standard development [2], [5], [81], and as part of an architecture of simple free-viewpoint television systems [4]. DSDE was already shown to provide better compression than depth coding with MV-HEVC, even when pixel rate constraint is not taken into account [15]. Moreover, the progressive development of edge-computing-based methods of view rendering for VR applications indicates that such scheme provides enough computational capabilities to perform multiview video processing and suitable network routing methods to ensure low latency [79], [80].

Since sufficient evidence has been presented to prove the relevance of this DSDE approach in immersive video coding [2], [4], [28], one of the MIV profiles implements this principle. The MIV Decoder-Side Depth Estimation (DSDE) using Geometry Absent profile allows MIV bitstreams to not include depth maps, keeping only the information required to reconstruct it. The main focus of this paper is to present the technical details of this novel profile and to propose extended solutions that further increase its usability and aid the depth estimation process. All the details of the MIV DSDE are presented in Section II.

The depth estimation in MIV is a non-normative part of the decoding process, therefore, any method can be used to compute depth maps. This paper consequently discusses in Section III-A a set of requirements for immersive video, particularly those that emerge from the unusual characteristics of the DSDE, that drives the selection of preferred depth estimation methods. Section III-B presents a summary of the literature related to the requirements of depth estimation for immersive video. Section III-C describes the depth estimation reference software which is collaboratively developed in parallel with the TMIV in order to tackle these requirements. Section IV presents the results of performed experiments and used test conditions (Section IV-B). In Section IV-C, the efficiency of the DSDE is evaluated compared to two other MIV profiles. This section includes as well the comparison of tested MIV profiles, summarized further in the form of a discussion on the advantages and disadvantages of the DSDE. Section IV-D explains how the DSDE can take advantage of a Geometry Assistance SEI message, and corresponding results are reported. By choosing a suitable depth estimation method it is possible to maximize the quality of the rendered video presented to the viewer. As far as we know, presented work reports results of the first evaluation of state-of-the-art multiview depth estimators in the DSDE context, their comprehensive experimental comparison is presented in Section IV-E. Section V concludes the paper and highlights some future possible work.

II. Decoder-Side Depth Estimation in MPEG Immersive Video

A. MIV Decoder Side Depth Estimation

The main MIV profile, also called Encoder-Side Depth Estimation (ESDE) in this paper, is described in detail in [1] and the corresponding reference test model, TMIV, is described in [10]. The DSDE mode is motivated by a variety of advantages over the ESDE mode. First of all, a significant amount of bitrate is saved, since the coded depth maps can represent more than 50% of the total bitstream in ESDE (especially for low bitrates – see experimental results in Table V). Simultaneously, twice the amount of pixel rate is saved and becomes advantageously available for textures, which in turn can be appropriately encoded with 2D codecs that are traditionally designed and optimized for texture rather than for depth content. While the 3D extension of HEVC enabled adequate compression of depths, this is no longer the case for video-based solutions like MIV. Consequently, the quality of light field reconstruction can be significantly higher in DSDE. Especially at high bitrates, the quality of estimated depth maps in DSDE is very close to the quality of depth maps obtained from uncompressed captured views.

While computational complexity of depth estimation can be expressed as a weakness of the DSDE mode, it is possible to significantly limit this complexity by inferring depth information only for the views contributing to the requested viewport. Without any additional feedback channel, this is impossible for the ESDE mode, for which estimating depth for all views at the encoder side is a burden for applications like dense light fields or for the streaming of live events.

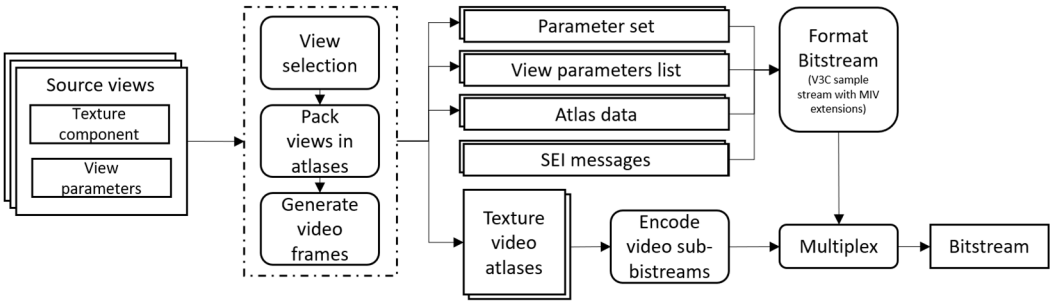


Fig. 1. Simplified TMIV encoder running in MIV DSDE mode.

The TMIV encoder is shown in Fig. 1, while the decoder and renderer are shown in Fig. 2. Both schemes have been simplified to reflect the relevant steps to run MIV in DSDE mode. The TMIV encoder selects a subset of the source views and packs their texture components into atlases [10]. This enables one decoder instantiation to process multiple views per video sub-bitstream. The view parameters consisting of camera intrinsic and extrinsic parameters, projection type, etc., together with views' position in the atlases (atlas data), and other crucial information for setting up the TMIV decoder such as atlas number and size, form the metadata that are encoded as a V3C [56] conformant bitstream. In general, the TMIV encoder is much simpler in DSDE mode, as pruning and depth processing are not performed.

The TMIV decoder parses the bitstream and provides the decoded texture atlases as well as the metadata to the renderer. Based on the requested viewport, the depth maps are estimated from the decoded views (available after reconstructing them from the decoded texture atlases). As the encoder ensures that pixel rate constraint is met in the encoded bitstream, the number of decoded views varies depending on their resolution. Nevertheless, the number of views that can be used for the estimation of depth is not restricted, therefore, all of them can be simultaneously used in this process. According to the depth maps and the view parameters, the views are projected and merged at the target view position (viewport). Holes in the synthesized view are filled through inpainting.

The algorithms used by the TMIV renderer are not normative. Currently, MPEG is using the Immersive Video Depth Estimation (IVDE) software (described in Section III-C) for the development of the DSDE mode. However, as we will show in this paper, any depth estimator may be used together with the MIV standard in order to provide high-quality immersive video.

B. Geometry Assistance SEI

One downside of the MIV DSDE mode is its inability to take advantage of high-quality depth maps which may be present at the encoder side. These depth maps may come along with computer-generated imagery (CGI) or may be tuned through manual or automatic refinement processes in case of natural content. To enable the use of these valuable data, the Geometry Assistance SEI message allows the transmission of certain side information or "features", which assist the depth estimator at the decoder side in computing higher quality depth maps with lower complexity. It is an adaptation of the Feature-Driven DSDE approach described in [15], but in MIV the block splitting is based on the cost volume (Section II-B-4), not on the sum of squared difference.

TABLE I
CODING OF POSSIBLE SPLIT TYPES IN GEOMETRY ASSISTANCE SEI

flag \ split type								
gas_split_flag	0	1	1	1	1	1	1	1
gas_quad_split_flag		1	0	0	0	0	0	0
gas_split_orientation_flag			0	0	0	1	1	1
gas_split_symmetry_flag			1	0	0	1	0	0
gas_split_first_block_bigger				0	1		0	1

It is a strict design philosophy of the MIV standard to keep the rendering a non-normative process, which also includes depth estimation. Therefore, these features cannot be derived from a specific depth estimation algorithm and must be reasonably universal. All features defined by the Geometry Assistance SEI can be directly extracted from the depth maps at the encoder side.

1) Partitioning: The motivation for the block-based nature of the SEI message is to adapt the features to the local properties of the depth maps. Initially, the depth map is divided equally into square blocks of size N. As a first feature, each block can be further divided into sub-blocks of square or rectangular shapes. The possible split types and their corresponding codes are shown in Table I.

As it can be seen, the signalization of the simplest square block is encoded just by one bit *gas_split_flag* code equal to zero, while to better adapt blocks to the local properties of more complicated structures present in depth maps, up to five bits are required. The bits have been assigned based on the occurrence of each split type in the reference implementation for the test sequences [78].

2) Depth Range: The second feature is the depth range, signaled for each of the blocks or sub-blocks. Z_{min} and Z_{max} are extracted from the depth map available at the encoder side. The depth range can be converted to disparities as $d_{min} = fb/Z_{max}$ and $d_{max} = fb/Z_{min}$ with the focal length f and the baseline b relative to a reference view. It enables the depth estimator at the decoder side to adapt the search interval $[d_{min}, \ldots, d_{max}]$ of disparity candidates for each of the blocks or sub-blocks. The most computationally complex step in

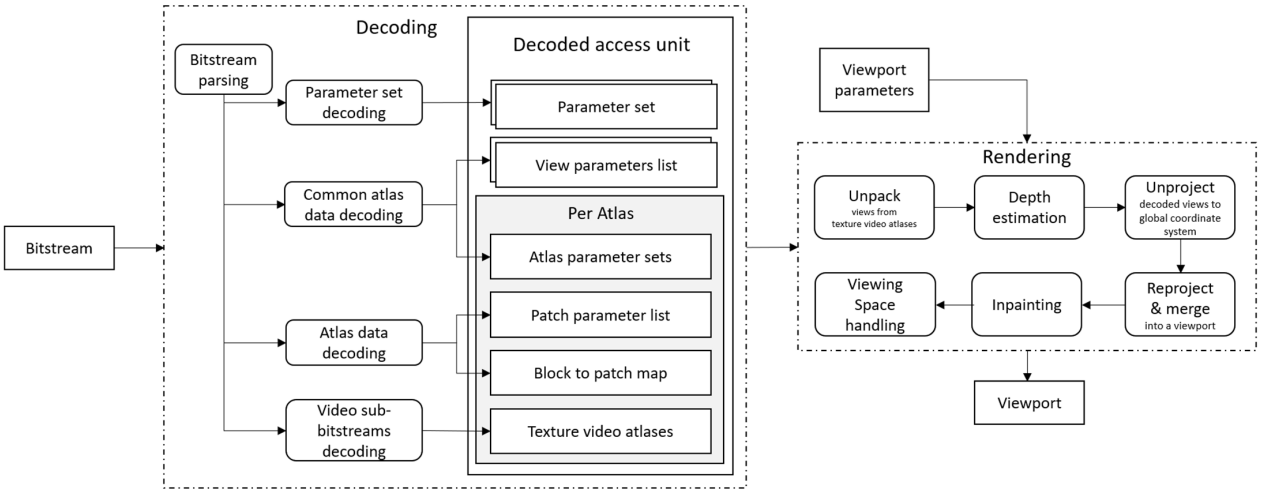


Fig. 2. Simplified TMIV decoder and renderer running in MIV DSDE mode.

depth estimation is the association of each depth candidate with a certain (dis-)similarity cost. In the case of an exhaustive search strategy, the total amount of disparity candidates to be tested is $M = (d_{max} - d_{min})\, p + 1$ with an optional sub-pixel precision p. Assuming these candidates are tested for each pixel, the resulting cost volume size is $CV = whM$, with the width w and the height h of the sequence. By restricting the search interval for each of the blocks and sub-blocks, the overall cost volume becomes a partial cost volume with a significantly smaller size.

The proposed approach yields two desirable consequences: first, the reduction of the cost volume translates into a reduction of the depth estimation complexity, as all typical depth estimation stages (*e.g.*, graph-cut [17], belief propagation,...) have a runtime that is roughly proportional to the cost volume size. Second, the removal of a large number of wrong disparity hypotheses prevents the depth estimator from erroneously selecting them and, therefore, increases the quality of the estimated depth map.

The MIV standard codes Z_{min} and Z_{max} in a predictive manner using the already coded values in the neighboring left $(Z_{min,left}, Z_{max,left})$ and top $(Z_{min,top}, Z_{max,top})$ block, indicated by *gas_ltmin_flag* and *gas_ltmax_flag* respectively. In that case, the depth ranges are derived as:

$$Z_{min} = (gas\_\text{ltmin\_flag} == 1 ? Z_{max,top} : Z_{max,left}) + gas\_qs * gas\_\text{zmin\_delta}$$
$$Z_{max} = (gas\_\text{ltmax\_flag} == 1 ? Z_{max,top} : Z_{max,left}) + gas\_qs * gas\_\text{zmax\_delta}$$

where the *gas_qs* value indicates the quantization step, while the differences between the current block and the indicated neighboring blocks depth range are *gas_zmin_delta* and *gas_zmax_delta*.

3) Depth Estimation Skip: The third feature is the depth estimation skip. It indicates for each block or sub-block that the depth estimation process may be entirely skipped, and that the depth information of the previous frame may be used instead. This flag has several advantages: firstly, the entire bitrate of the Geometry Assistance SEI is greatly reduced, since the depth ranges are not coded if a block or sub-block is indicated as depth estimation skip. Secondly, if the flag is considered by the depth estimation process, robust temporal stability can be achieved in the depth maps and therefore in the synthesized view. Finally, the depth estimation skip reduces the complexity of the entire depth estimation step for a future frame.

In the MIV standard, the depth estimation skip is indicated by the *gas_skip_flag*. If it is equal to zero, *gas_zmin_delta* and *gas_zmax_delta* have to be present in the bitstream, while *gas_ltmin_flag* and *gas_ltmax_flag* may be present. If the skip flag is equal to one, none of these syntax elements are present in the bitstream.

4) Feature Extractor: The TMIV encoder includes a feature extractor, which implements one possible derivation of the three features from the available depth maps. Two thresholds $T_{skip}$and T_{split} are defined. The partitioning is performed if the amount of cost volume reduction is below T_{split}: $CV_N / CV_{split} < T_{split}$, considering the size of initial cost volume CV_N and the size of cost volume of the split block CV_{split}. A depth estimation skip is performed for a block or sub-block if the L_1 distance between the current and the previous depth block or sub-block is below T_{skip}. If a block or sub-block has a depth estimation skip flag set, the corresponding cost volume is assumed to be zero.

5) Additional Signaling and Implementation Details: We provide the remaining details on the Geometry Assistance SEI signaling in the MIV standard. Besides elements described above, and before a set of cascading flags to indicate block division types, two fields indicate the initial size of the undivided blocks, and the quantization step of the depth range values.

To minimize the size of encoded metadata, when a minimum and maximum depth is transmitted for a block, they are first predicted from the depth range values of the top or left block, the choice between those two predictors being signaled by a flag at the block level. The quantized residual is signaled for the minimum and maximum depth values.

III. Depth Estimator for DSDE

A. Requirements for a Decoder-Side Depth Estimator

This section provides a set of requirements that preferably have to be met by depth estimation techniques at the

decoder side in the context of immersive video. Some of these requirements are common to ESDE and DSDE, while others emerge from specific challenges raised by the DSDE framework. In the following, we first summarize the common and then the DSDE-specific requirements.

Currently, the TMIV provides support for perspective, equirectangular (ERP), and orthographic projection of the source views. Therefore, depth estimation methods used for the development of standards have to be as versatile as possible to support these formats. Typically, the views are of very high-resolution and frame rates, in order to maintain the feeling of immersion for the end user [47].

As immersive video targets light fields sampled from multiple cameras, the depth estimator should also be able to handle multiple input views and achieve inter-view consistency of the depth maps [3], [21]. Furthermore, to provide convincing view synthesis, the quality of the depth maps needs to be high, *i.e.*, the object boundaries need to be respected in order to avoid a mixture between foreground and background. Also, flexibility towards the camera setup and sparsity is required. Possible camera arrangements vary significantly in multi-camera systems, from one-dimensional linear [41], or two-dimensional matrix-like arrangements [36], to the spherical outward-looking placement of cameras [35].

In the context of ESDE, it can be assumed that the fully sampled light field is available and that the views are uncompressed. However, in DSDE this assumption is not correct. First of all, a depth estimator must deal with decoded input views, which may be severely distorted by the compression. Thus, besides the challenges typically faced by the image-based depth estimation, like texture-less regions, occlusions, or specular reflections [67], the compression-induced errors can affect the robustness of inter-view matching. During the video compression, a large part of high-frequency components can be notched out, which in turn leads to blurring or, in the worst case, an edge shift in the image. At the same time, the number of views may be significantly sparser than in the ESDE case, which may make it difficult to find matches at all. Consequently, in rare cases, a depth estimator may have to support monocular depth estimation as well.

Furthermore, in a situation where patch atlases are used in the DSDE context [28], a depth estimator may have to deal with small patches and significantly less information. Given that such an approach can be quite complex and decreases the number of applicable depth estimation methods, the current DSDE framework is only utilizing full frames.

Finally, the DSDE context comes with strong constraints on complexity. In order to achieve 30 to 60 fps per view with input views not smaller than full HD (or even 4K to provide satisfactory level of immersion for a viewer [85]), compromises in terms of performance may be required, as meeting all previously listed requirements is challenging.

B. Depth Estimation for Immersive Video

The literature often provides depth estimation solutions for only a subset of the listed requirements. For example, wide support for all required projections is not common, as most of the work in depth estimation is focusing on perspective binocular stereo rigs [20]. Depth estimation from ERP views is also commonly researched [29], [30], [31], but both projections are rarely supported simultaneously, especially for deep learning methods where learning from both perspective and omnidirectional images does not provide satisfactory quality [72]. Nevertheless, since real-world multimedia systems are usually designed for a particular application, wide support for different types of input views is not so crucial outside the standardization process.

In terms of view arrangement, besides the mentioned linear stereo rigs, the commonly used light-field-like camera setup is often supported [22], [54], even for unrectified content [23]. The wide-baseline arrangement is also supported in some methods [24], which is significant for immersive video applications, as such arrangement provides a larger volume for possible viewports, increasing the user's immersion into the watched video.

Inter-view consistency among the depth maps can be recovered through post-processing of the depth maps [19]. However, such an approach can also be a part of the estimator itself. For instance, MVSNet [46] successfully handles depth estimation for multiple input views. Instead of computing the whole geometry of the scene, it computes only one depth map at a time, while applying a simple depth residual learning network as a post-processing step. Extensions of this method are continuously proposed to provide unsupervised dense point cloud reconstruction, as described in [55].

The high-resolution images still pose a major challenge for machine-learning-based depth estimators [18], [20]. However, the research on depth super-resolution is well developed due to its usability in depth (time-of-flight) camera applications [66].

Real-time depth estimators are often motivated by other applications like robotics [25], automated driving [64], or low-resolution estimation for augmented reality in mobile devices [63], therefore, do not yet provide sufficient performance for immersive video. Yet, deep-learning based monocular depth estimation is also progressing quickly [62], [65], indicating the possibility of using them in near future also for immersive video applications.

While the resolution of input views mainly affects the time required for the depth estimation, the domain shift problem can directly degrade the quality of depth maps [18]. The amount of required ground truth samples that will allow to properly model both the characteristics of CGI and natural content that, besides, are compressed in the case of DSDE, is rising. Hence, the deployment of a selected end-to-end network that will handle such diversified content is a very challenging task.

Research on the influence of compression-related errors on the quality of estimated depth is limited and narrowed to monocular depth estimation [73]. Therefore, to show if a reasonable rendered view quality can be achieved with machine-learning-based multiview depth estimation without retraining of the network, experiments on using different depth estimators on the decoder-side are presented in Section IV-E. Tested methods are GANet [14] and GWCNet [6]. The GANet is a method that improves the accuracy of the depth estimation in some particularly challenging areas, such as occlusions,

reflective regions, and thin structures. The GWCNet provides efficient representations to compute feature similarities and decreases the inference computational cost.

While deep learning methods for depth estimation constitute a major direction for research, the use of conventional methods remains very convenient in the development stage of the standardization process, as domain shift problems that could occur for deep learning-based methods are eliminated, decreasing the overall complexity of conducted experiments, as retraining is not required. The most widely known methods are based on optimization using Markov Random Fields (MRF) [17] and often provide very high-quality estimated depth maps, which is shown in the rankings of the Middlebury database and benchmark [68]. Few of the most accurate methods utilize MRF with segments / patch matching, *e.g.*, [69]–[71]. Such high quality and versatility come at the cost of greater computational complexity than in the case of some deep learning methods.

C. Immersive Video Depth Estimation Software

The MPEG reference software for depth estimation, called Immersive Video Depth Estimation (IVDE) [9], meets the above-listed requirements. It addresses depth estimation from videos acquired by any number of arbitrarily positioned omnidirectional or perspective cameras.

Similarly to its widely known predecessor, called Depth Estimation Reference Software (DERS) [13], [16], IVDE is based on the minimization of a cost function that utilizes an MRF graph [17]. The core of the IVDE algorithm is based on a graph-based method described in [3]. Depth is estimated for segments instead of individual pixels, and thus the size of segments can be adjusted to control the trade-off between the quality of the depth maps and the processing time. Larger segments can be used to achieve faster depth estimation, while finer segments can be used to achieve higher quality. Object boundaries are usually closely collocated with segment borders, therefore segment-based depth estimation does not reduce the depth map precision. Such an approach decreases the negative influence of the high resolutions of source views on the complexity of the estimation; the number of segments (calculated using the superpixel segmentation method from [8]) can be fixed, regardless of the resolution. Simultaneously, using views with higher resolutions increases the quality of depth maps, as the estimated depth for each segment is calculated on a per-pixel basis.

The estimation is performed for all views simultaneously and it is inter-view consistent thanks to the formulation of the cost function, dedicated for segment-based estimation. The depth maps estimated in previous frames may be utilized in the estimation of depth for the current frame, producing temporally consistent depths, thus decreasing the processing time.

The combination of these features makes IVDE well-adjusted for immersive video applications, and currently, five out of six natural content multiview sequences in Common Test Conditions for MPEG Immersive Video use depth maps estimated by IVDE [11]. In addition, it is continuously and collectively updated to closely match the DSDE requirements. Some of the improvements that were lately introduced include a new matching method adapted for compressed input views (the point-to-block matching [7]), an automatic calculation of the depth range [9], an adaptation of the superpixel segmentation for omnidirectional views [9], and the support of Geometry Assistance SEI (described in Section II-B).

IV. Experimental Results

A. Overview

This section provides the results of experiments conducted to comprehensively test the DSDE mode in TMIV. After presenting the test environment, we present the results of three experiments:

1. The comparison between the ESDE and DSDE in MIV (Section IV-C).
2. The enhancement of the DSDE utilizing the Geometry Assistance SEI message of MIV (Section IV-D).
3. The performance of the MIV DSDE with different state-of-the art depth estimators (Section IV-E).

B. Common Test Conditions

In order to conduct fair comparisons between different experimental schemes, common test conditions are essential. The MPEG Video Coding group defines such common test conditions [11] for assessing different competing proposals. We strictly follow these test conditions in all our experiments.

The TMIV 9 reference software [10] processes input views and depths (when present) to produce atlases and corresponding metadata. It compresses and decompresses these atlases using VVenC and VVdeC [50], a fast implementation of VVC. It renders final output views in the same position as the input views. In DSDE mode, depth estimation is initiated prior to view synthesis. The following "pixel-rate" constraints are imposed on all configurations of TMIV:

- The combined luma sample rate across all decoders does not exceed 1069547520 samples per second (as in HEVC Main10 profile level 5.2).
- Each coded video picture size does not exceed 8912896 pixels (*i.e.*, 4096×2048).
- The number of decoder instantiations does not exceed 4.

The performance is assessed through the quality of rendered views. Two full-reference objective video quality metrics are used: the Weighted-to-Spherically-uniform Peak Signal-to-Noise Ratio (WS-PSNR) [51] and the Immersive Video PSNR (IV-PSNR) [32]. The PSNR is the most used metric to quantify reconstructed video quality (here it is calculated for the luma and used in the form adapted also for omnidirectional video). IV-PSNR is a metric specifically designed to reflect virtual view synthesis artifacts (calculated for luma and chroma jointly). The metrics are applied on the luminance component of the rendered view.

Five different rate points (RP) are used as listed in the CTC [11]. The Bjøntegaard delta [33] (which shows the percentage change in the bitrate required to achieve the same quality for two measured coding techniques) is calculated for each metric, for the four smallest QPs (high bitrates)

TABLE II
LIST OF TEST SEQUENCES

Sequence	Source	Type		Resolution	Views
ClassroomVideo	[34]	ERP	CG	4096 × 2048	15
Chess	[35]	ERP	CG	2048 × 2048	10
Hijack	[36]	ERP	CG	4096 × 2048	10
Museum	[36]	ERP	CG	2048 × 2048	24
Group	[38]	Perspective, convergent	NC	1920 × 1080	21
Fencing	[42]	Perspective, convergent	NC	1920 × 1080	10
Fan	[37]	Perspective, planar	NC	1920 × 1080	15
Kitchen	[39]	Perspective, planar	NC	1920 × 1080	25
Mirror	[40]	Perspective, planar	NC	1920 × 1080	15
Carpark	[44]	Perspective, planar	NC	1920 × 1088	9
Frog	[43]	Perspective, planar	NC	1920 × 1080	13
Hall	[44]	Perspective, planar	NC	1920 × 1088	9
Street	[44]	Perspective, planar	NC	1920 × 1088	9
Painter	[45]	Perspective, planar	NC	2048 × 1088	16

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

and for the four largest ones (low bitrates). Consequently, we present high-bitrate as well as low-bitrate BD-Rates (high-BR BD-Rates and low-BR BD-Rate) in each table together with the atlas encoding, video encoding as well as atlas decoding and rendering runtimes. In cases of insufficient overlap, the BD-Rate may not be computable. We indicate these cases with a "—" in the tables and provide additional information in-text. Additionally, a gain or a loss is always indicated by a green or red cell respectively.

A set of multiview test sequences is used. The test set covers not only natural content (NC), but also CGI content, both having depth maps estimated using IVDE, in order to provide a fair comparison between ESDE and DSDE. The sequences were acquired with planar or convergent rigs equipped with various number of cameras. A brief summary of test sequences is available in Table II. In all experiments 17 frames are used for the evaluation. Besides in listed sources, a subset of test sequences is available publicly on the MPEG MIV website [86].

The CTC defines the three anchors of the TMIV. The MIV Atlas and the MIV View anchors belong to the MIV ESDE, while the MIV Decoder-Side Depth Estimation anchor belongs to the MIV DSDE. Each anchor describes a different strategy of utilizing the TMIV. In the MIV Atlas anchor, the TMIV encoder first labels each view as either "basic" or "additional". Then, the encoding preserves the basic views and subdivides the additional views into patches that only contain non-redundant data deviating from the basic views' field of view [1]. In the MIV View anchor, the TMIV encoder first selects and encodes a greater number of basic views without considering any additional view. Consequently, the atlases of the MIV View anchor contain only full views and no patches.

C. Comparison of MPEG Immersive Video Profiles

In this section, we present the performance of the DSDE compared to the two anchors of the MIV ESDE. Since both ESDE anchors require depth information at the encoder side, they serve as good comparison targets for verifying the merits of using DSDE.

1) Results: We strictly follow the CTC described in Section IV-B. The results of the comparison of ESDE and DSDE in MIV are presented in Tables III and IV. They indicate significant BD-Rate gains (represented by green cells, negative percentage indicates reduction of bitrate required to achieve the same quality) for the MIV DSDE anchor for the majority of sequences compared to either MIV Atlas or MIV View. For many sequences, the bitrate is reduced by more than 50% (an example of BD-Rate curve for one of these sequences is shown in Fig. 3a). This result is a combination of omitting the depth transmission and replacing it with efficient coding of textures. For three sequences the quality in DSDE is much better only for low bitrates (*e.g.* Carpark – Fig. 3b). Just few sequences show degradation of quality, *e.g.,* Chess (Fig. 3c), or Hall (Fig. 3d). The first one is a very challenging multiview sequence with glossy textures that hinder depth estimation process. The latter, as it can be seen in Fig. 3d, still provides the high quality for both ESDE and DSDE.

The gain in compression performance is naturally tied with moving the complexity from the encoder side to the decoder side. As explained in Section II-A, the runtimes reported in

TABLE III
BD-RATES SAVINGS AND RUNTIMES CHANGES OF ENCODING AND DECODING OF MIV DSDE OVER MIV ATLAS

Sequence	High-BR BD rate WS-PSNR	Low-BR BD rate WS-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Atlas encoding	Video encoding	Decoding & Rendering
ClassroomVideo	-73.4%	-82.3%	-67.0%	-82.2%	0.4%	53.4%	955.5%
Museum	---	---	-87.5%	-85.6%	0.1%	42.7%	918.1%
Fan	-53.9%	-67.4%	-42.2%	-57.2%	1.0%	80.6%	2929.3%
Kitchen	-61.4%	-55.5%	-11.4%	-22.0%	0.6%	60.8%	2687.8%
Painter	-73.8%	-73.1%	-60.7%	-66.2%	1.2%	74.8%	4322.0%
Frog	-63.7%	-60.7%	-51.2%	-55.9%	1.2%	85.6%	7297.2%
Carpark	-0.0%	-28.3%	-2.9%	-27.5%	1.7%	83.2%	2689.7%
Chess	---	---	---	---	0.3%	62.4%	1837.0%
Group	---	---	---	---	0.3%	73.1%	3266.1%
Fencing	39.6%	-22.8%	18.1%	-39.2%	1.6%	107.1%	2529.2%
Hall	1934.4%	739.9%	1249.8%	485.7%	1.6%	204.3%	1956.8%
Street	-68.3%	-55.6%	-38.6%	-42.0%	2.0%	85.7%	2967.2%
ChessPieces	---	---	---	---	0.4%	53.0%	1931.4%
Hijack	---	---	---	---	0.3%	67.8%	506.5%
Mirror	-14.4%	-42.1%	18.2%	-32.9%	1.0%	80.8%	2926.0%

TABLE IV
BD-RATES SAVINGS AND RUNTIMES CHANGES OF ENCODING AND DECODING OF MIV DSDE OVER MIV VIEW

Sequence	High-BR BD rate WS-PSNR	Low-BR BD rate WS-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Atlas encoding	Video encoding	Decoding & Rendering
ClassroomVideo	---	-96.6%	-63.9%	-71.5%	2.0%	57.3%	3018.3%
Museum	---	-91.6%	-75.6%	-70.9%	2.1%	36.4%	3887.4%
Fan	---	-83.8%	---	-77.7%	5.3%	119.0%	5179.2%
Kitchen	---	-87.3%	-86.4%	-64.5%	4.8%	77.8%	5728.7%
Painter	---	-75.3%	-82.2%	-73.5%	4.5%	77.7%	6216.1%
Frog	-54.7%	-52.2%	-56.9%	-53.8%	4.9%	116.3%	10342.8%
Carpark	7.6%	-22.8%	1.5%	-23.5%	4.7%	85.0%	3000.8%
Chess	---	---	---	---	5.6%	54.8%	3777.2%
Group	---	---	---	---	3.2%	84.4%	5854.6%
Fencing	-19.3%	-44.9%	-52.1%	-60.7%	4.4%	111.9%	3605.2%
Hall	1565.8%	2056.8%	949.1%	662.7%	6.5%	202.9%	2281.1%
Street	-22.7%	-32.8%	-33.8%	-42.0%	5.0%	82.8%	3285.9%
ChessPieces	---	---	---	---	4.8%	31.8%	3481.2%
Hijack	---	---	---	---	3.0%	88.7%	1357.3%
Mirror	-62.7%	-52.8%	-58.5%	-49.2%	6.6%	70.7%	4139.3%

TABLE V
THE AVERAGE DISTRIBUTION OF BITRATE PER DATA TYPE IN MIV ATLAS

Test point	Average bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	49.412	18.556	0.131	68.099	71.5%	28.3%	0.2%
RP2	24.036	13.551	0.131	37.717	61.1%	38.5%	0.4%
RP3	10.960	9.043	0.131	20.133	52.5%	46.8%	0.7%
RP4	4.855	5.444	0.131	10.430	44.7%	53.9%	1.3%
RP5	2.164	2.810	0.131	5.104	40.1%	57.2%	2.7%

TABLE VI
THE AVERAGE DISTRIBUTION OF BITRATE PER DATA TYPE IN MIV DSDE

Test point	Average bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	36.862	0	0.009	36.870	100.0%	0%	0.0%
RP2	21.324	0	0.009	21.332	99.9%	0%	0.1%
RP3	11.635	0	0.009	11.644	99.9%	0%	0.1%
RP4	5.962	0	0.009	5.970	99.8%	0%	0.2%
RP5	3.031	0	0.009	3.039	99.6%	0%	0.4%

these tables do not take advantage of the ability of DSDE to consider the known requested viewport, by computing only the required depth maps, as any practical implementation of DSDE would do. Instead, depth maps are estimated for all views. However, when encoding atlases, the MIV DSDE mode is much faster. This is a considerable advantage, keeping in mind that for the ESDE modes, the time required for depth estimation is not part of the reported encoder complexity, as it is considered that depth maps are available prior to coding. The simplicity of the TMIV encoder in DSDE mode is also reflected in the block diagram in Fig. 1.

On average, DSDE provides higher gains for low bitrates. This is highly linked with the average bitrate that is required for depth encoding in ESDE modes. As shown in Table V, the bitrate of the depth is a significant part of the overall bitstream: it ranges from 28% of the total bitrate for high qualities to 57% for low ones. Depth maps are highly prone to compression artefacts when encoded with general-purpose 2D video codecs, therefore, using high compression on depth maps results in a large decrease in the synthesized views quality. The avoidance of this problem is a huge advantage of the DSDE mode in MIV. Furthermore, the MIV DSDE anchor does not require a selection of appropriate quantization parameters for the depth, which is often non-trivial to maintain the correct ratio between texture and depths.

The distribution of the bitrate for MIV DSDE is presented in Table VI. Almost all bits are spent for the texture, as for low bitrates only 0.4% of the total bitrate is spent for the MIV metadata. Therefore, assuming that the depth estimated at the decoder side will not be of much worse quality than the one available in the encoder, the textures used for the synthesis will be of better quality with DSDE than with ESDE at similar bitrates.

As objective results might sometimes be misleading for evaluating synthesized views, we provide some visual comparisons of rendered viewports for selected sequences in Fig. 4. The views are rendered between positions of input views, with bitrates smaller than 10 Mbps and closely matching for both compared modes. As it can be observed, this visual comparison confirms that for such a low bitrate, MIV DSDE can provide a much larger level of detail when compared to the state-of-the-art in immersive video compression, MIV Atlas.

Much sharper texture can be seen in all presented views, even for the Chess test sequence, where DSDE shows objective loss when compared to ESDE (Table IV). The loss is the result of depth estimation errors that can be observed in the synthesized viewport on the edges of chess pieces and the non-Lambertian surface of the chessboard. The estimation of depth on such challenging content is still a very demanding task (as underlined also by authors of the latest surveys of state-of-the-art techniques [18], [20]), nevertheless, MIV DSDE can support any depth estimator, so incoming novel methods that will show improvement in these fields will ensure even better quality for the viewer without changing any other parts of the framework.

D. Geometry Assistance SEI

1) Methodology and Design of the Experiments: The DSDE mode provides optional Geometry Assistance SEI. A comparison of DSDE with and without the SEI message is provided in this section. The experiment follows the methodology from Section IV-B, adding the use of features extracted from depth maps at the encoder side (as presented in Section II-B), using IVDE depth maps. Based on preliminary tests, we set $T_{skip} = 0.2\%$ and $T_{split} = 4\%$. The overall bitrate of features is limited to not exceed 1 Mbps, which is achieved by adapting the strength of their quantization.

2) Results: Table VII shows the results comparing MIV DSDE and MIV DSDE with Geometry Assistance SEI. The primary goal of the Geometry Assistance SEI message is the reduction of the decoder-side complexity. Consequently, the estimation of depth is speeded up by more than two for most sequences. Simultaneously, the compromise in BD-Rate performance is varying over the sequences, but similar on average. In several cases, significant quality improvements of over 1 dB are measured, due to the avoidance of testing disparity candidates, which lie outside the correct range. The highest increase in BD-Rate can be observed for the sequences that performed the worst in comparison shown in Table III, which indicates that the use of Geometry Assistance SEI is particularly advantageous for such challenging sequences with three-dimensional geometry that is difficult to estimate on the decoder side. These results indicate, that the Geometry Assistance SEI message can support the depth estimator in providing higher quality depth maps while simultaneously reducing the complexity. This property is crucial, as typically, depth estimators sacrifice accuracy in order to speed up the depth estimation process.

It can be seen as a possible disadvantage of the Geometry Assistance SEI, that depth maps must be present at the encoder side, from which the features are extracted. The higher the quality of the depth maps, the more accurate the features and the better the performance of the depth estimator at the decoder side [15]. However, if high quality depth maps are present at the encoder side, should one utilize the ESDE or the DSDE with Geometry Assistance SEI? In order to answer

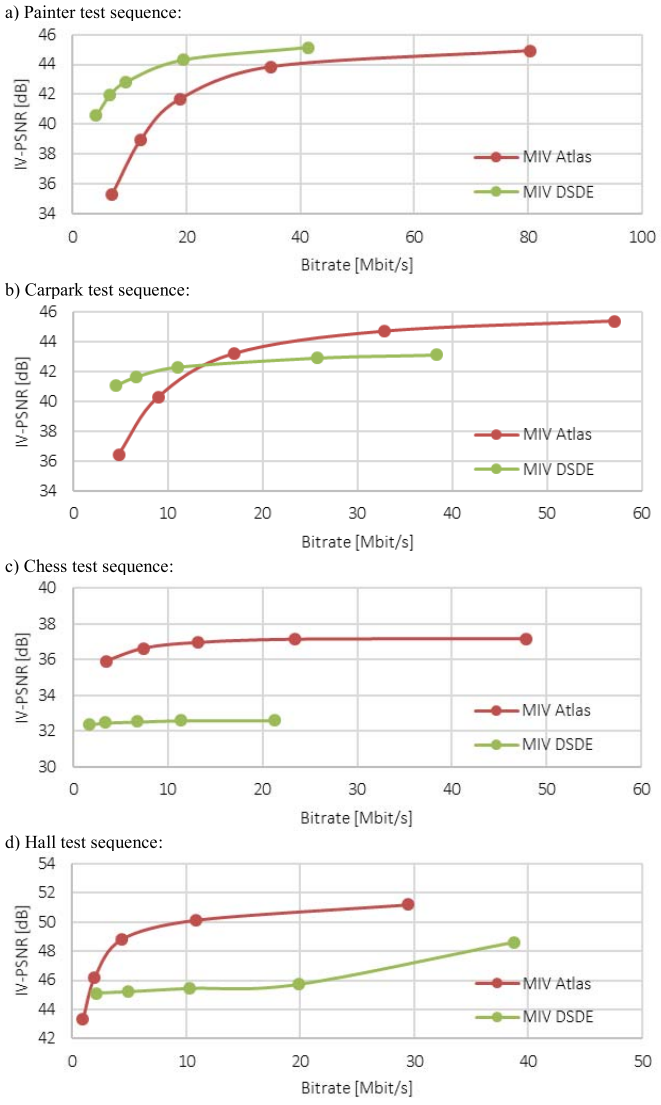


Fig. 3. IV-PSNR BD-Rate curves for three observed scenarios: a) MIV DSDE is better than MIV Atlas for all rate points (in 8 out of 15 sequences, shown here on the example of painter), b) MIV DSDE is better for low bitrates (3 sequences—Carpark is presented), c) and d) MIV DSDE is worse for all rate points (4 sequences—Chess and Hall are presented).

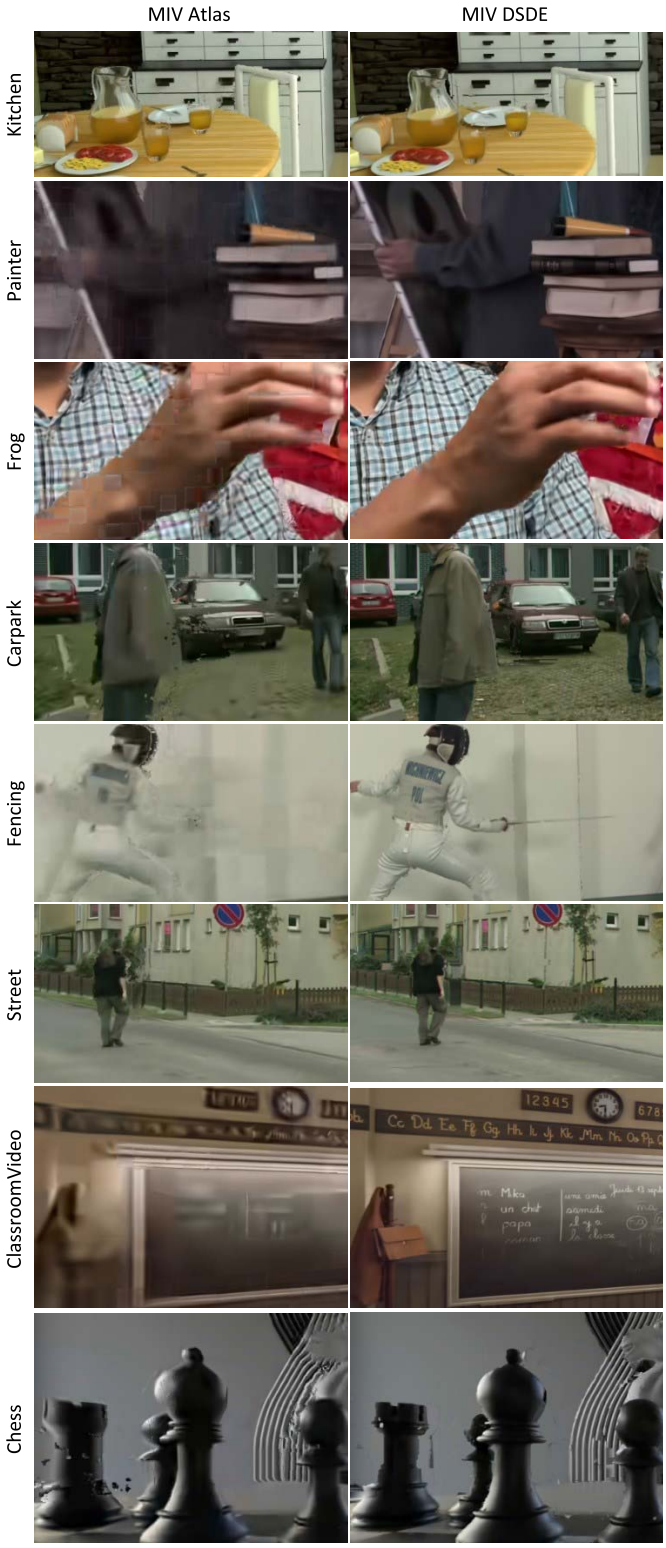


Fig. 4. The subjective comparison of MIV Atlas (left) and MIV DSDE (right) for fragments of selected viewport synthesized between positions of input views; the bitrate for all cases was smaller than 10 Mbit/s and closely matching.

this question, we additionally provide a comparison of DSDE with Geometry Assistance SEI and the MIV Atlas anchor in Table VIII.

In this case MIV DSDE still shows gains in comparison with the atlas-based encoding, but the decoding and rendering time becomes more competitive, as for some sequences it is reported to be only three times slower (ClassroomVideo, Hijack). We can therefore conclude, that the DSDE with Geometry Assistance SEI is a superior solution over tested ESDE profiles in terms of BD-Rate. While complexity concerns may still be an argument in favor to the ESDE profile, the Geometry Assistance SEI has shown to make a huge step towards reducing the complexity concern.

Another interesting analysis is the comparison between the bitrate cost of depth maps versus the Geometry Assistance SEI message. Table IX presents the bitrate invested into the geometry Assistance SEI message for all rate points (RP). Similarly to the DSDE anchor, most of available bitrate is spent on the texture, but the amount of metadata is increased by the cost of the Geometry Assistance SEI. Nevertheless, this

TABLE VII

BD-RATES SAVINGS AND RUNTIMES CHANGES OF ENCODING AND DECODING OF MIV DSDE WITH GEOMETRY ASSISTANCE SEI OVER MIV DSDE

Sequence	High-BR BD rate WS-PSNR	Low-BR BD rate WS-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Atlas encoding	Video encoding	Decoding & Rendering
ClassroomVideo	83.2%	95.5%	19.0%	45.1%	100.0%	100.0%	35.0%
Museum	133.2%	105.8%	74.3%	87.0%	100.0%	100.0%	65.1%
Fan	5.9%	9.4%	8.9%	13.2%	100.0%	100.0%	27.7%
Kitchen	25.5%	20.5%	26.8%	22.0%	100.0%	100.0%	46.3%
Painter	-13.9%	-0.5%	-7.5%	2.3%	100.0%	100.0%	43.0%
Frog	5.3%	7.9%	20.2%	16.6%	100.0%	100.0%	16.7%
Carpark	-75.8%	-52.2%	-70.6%	-51.3%	100.0%	100.0%	44.3%
Chess	---	---	-56.3%	-57.0%	100.0%	100.0%	50.8%
Group	---	---	---	---	100.0%	100.0%	44.4%
Fencing	-29.9%	-10.7%	3.1%	8.8%	100.0%	100.0%	24.3%
Hall	---	---	-86.1%	---	100.0%	100.0%	41.5%
Street	14.5%	12.0%	5.5%	8.5%	100.0%	100.0%	34.0%
Hijack	---	---	---	---	100.0%	100.0%	48.5%
Mirror	-13.1%	0.3%	-22.9%	-4.3%	100.0%	100.0%	43.8%

TABLE VIII

BD-RATES SAVINGS AND RUNTIMES CHANGES OF ENCODING AND DECODING OF MIV DSDE WITH GEOMETRY ASSISTANCE SEI OVER MIV ATLAS

Sequence	High-BR BD rate WS-PSNR	Low-BR BD rate WS-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Atlas encoding	Video encoding	Decoding & Rendering
ClassroomVideo	-48.9%	-68.2%	-64.2%	-74.4%	0.4%	53.4%	334.7%
Museum	---	---	-76.8%	-71.3%	0.1%	42.7%	597.9%
Fan	-52.0%	-64.0%	-38.7%	-51.6%	1.0%	80.6%	812.7%
Kitchen	-51.0%	-46.8%	14.2%	-5.0%	0.6%	60.8%	1244.0%
Painter	-75.3%	-71.3%	-62.6%	-64.4%	1.2%	74.8%	1858.5%
Frog	-61.3%	-57.1%	-42.2%	-49.0%	1.2%	85.6%	1220.5%
Carpark	-30.3%	-42.4%	-31.6%	-42.7%	1.7%	83.2%	1192.7%
Chess	---	---	---	---	0.3%	62.4%	933.2%
Group	---	---	---	---	0.3%	73.1%	1449.0%
Fencing	69.3%	-19.6%	28.0%	-32.7%	1.6%	107.1%	614.9%
Hall	644.0%	133.2%	174.8%	54.6%	1.6%	204.3%	811.6%
Street	-63.8%	-51.0%	-35.3%	-36.9%	2.0%	85.7%	1010.0%
Hijack	---	---	---	---	0.3%	67.8%	245.8%
Mirror	-22.6%	-39.2%	-2.2%	-30.2%	1.0%	80.8%	1280.3%

TABLE IX

THE AVERAGE DISTRIBUTION OF BITRATE PER DATA TYPE IN MIV DSDE WITH GEOMETRY ASSISTANCE SEI

Test point	Bitrate [Mbps]				Fraction [%]		
	Texture	Depth	Metadata	Total	Texture	Depth	Metadata
RP1	36.862	---	0.712	37.574	97.8%	0.0%	2.2%
RP2	21.324	---	0.712	22.036	95.6%	0.0%	4.4%
RP3	11.635	---	0.712	12.348	92.3%	0.0%	7.7%
RP4	5.962	---	0.712	6.674	87.2%	0.0%	12.8%
RP5	3.031	---	0.712	3.743	78.8%	0.0%	21.2%

bitrate remains significantly lower than the bitrate required for compressed depth maps in MIV Atlas (Table V), making the proposed SEI a very valuable tool for efficient encoding of immersive videos and a promising alternative to classical video-based depth coding.

E. Depth Estimation

The current MIV DSDE adopts IVDE as the reference tool for estimating depth maps at the decoder side. However, to enable a variety of different applications and scenarios, the DSDE is designed as compatible with a broad range of depth estimators. This flexibility is verified by replacing IVDE in the DSDE workflow with several well-known depth estimation techniques. They are carefully chosen to cover not only the traditional methods heavily based on multiview depth estimation but also the emerging data-driven deep learning techniques. The comparison of these estimators is presented in this section.

1) Methodology and Design of the Experiments: In this section, the bitrates are identical as in the experiments presented in Section IV-C, because the only change in the framework is the depth estimator used at the decoder side. Therefore, a direct comparison of quality of rendered views can be performed, making it possible to focus only on the differences in the quality introduced by different depth estimation methods. To present a wide comparison that takes into account different type of distortions that can be observed for each tested case, additional robust machine-learning-based quality metrics were utilized: LPIPS [52] and VMAF [53].

The examined candidates include DERS [16], GANet [14] and GWCNet [6]. DERS has been developed by the MPEG community as a combined effort of multiple organizations. Its technical maturity has been proved in the academic field, as it is widely used as the reference technique to evaluate new depth estimation approaches. In addition, DERS is designed to be applicable for immersive video scenarios by not imposing any restrictions on camera structures. Therefore, DERS can be considered as a proper reference for the presented framework.

GANet and GWCNet are state-of-the-art end-to-end stereo matching networks, based on cost volume matching with 3D convolution. These methods were chosen as one of the highly recognized methods in the literature. The experiment evaluates whether these methods can be efficiently utilized in the MIV context without performing any retraining. For both methods, we used the default models pre-trained on the KITTI dataset [25], [26]. Only perspective and rectified content is included in this analysis to ensure fair testing conditions for deep learning approaches, trained by default on such type of test sequences. The disparity maps produced by GANet and GWCNet are subsequently converted into depth maps [27].

2) Results: The quality of the synthesis for five different rates for all tested depth estimation methods is presented in Table X. In the objective comparison using PSNR and IV-PSNR, the best average results can be observed for DERS, but the results vary for different sequences, as roughly for half of the sequences the best quality is either for DERS or for IVDE. Both methods propose similar approach of estimating depth as the optimization of energy function based on the graph structure, but differ significantly on many technical bases, such as estimation unit, which in IVDE is a superpixel, while DERS performs pixel-wise depth estimation. This results in better pixel-to-pixel correlation of synthesized view and the input view, represented as higher quality in PSNR-based methods. For VMAF and LPIPS, which were shown to provide high correlation with subjective quality, the average results show that for low bitrates the best quality is provided by

TABLE X
THE COMPARISON OF AVERAGE QUALITY OF RENDERED VIEWS FOR DIFFERENT DEPTH ESTIMATION METHODS USED IN MIV DSDE

	Rate	WS-PSNR [dB]				IV-PSNR [dB]				VMAF				LPIPS			
		IVDE	DERS	GANet	GWCNet	IVDE	DERS	GANet	GWCNet	IVDE	DERS	GANet	GWCNet	IVDE	DERS	GANet	GWCNet
Fan	RP1	32.48	**32.82**	29.42	29.58	40.51	**40.90**	36.96	37.12	84.71	**85.89**	72.42	74.09	0.085	**0.084**	0.094	0.094
	RP2	32.30	**32.66**	29.35	29.49	40.32	**40.66**	36.86	37.02	84.14	**85.75**	71.98	73.56	**0.089**	0.090	0.099	0.098
	RP3	31.75	**32.11**	29.05	29.13	39.62	**40.02**	36.43	36.58	82.12	**83.10**	70.08	71.46	**0.102**	0.105	0.112	0.111
	RP4	30.37	**30.51**	28.18	28.15	37.91	**38.00**	35.25	35.33	75.61	**76.12**	64.34	65.17	**0.132**	0.138	0.144	0.142
	RP5	**28.22**	28.06	26.56	26.02	**35.18**	35.05	33.07	32.26	62.15	**61.70**	52.92	52.91	**0.166**	0.176	0.179	0.178
Kitchen	RP1	**35.52**	35.26	31.48	29.18	**43.78**	41.70	39.12	36.57	**88.81**	89.76	80.53	76.28	**0.162**	0.163	0.167	0.169
	RP2	**34.93**	33.95	31.25	29.04	**43.03**	40.36	38.83	36.36	87.79	**88.03**	79.50	75.65	**0.190**	0.192	0.193	0.194
	RP3	**34.13**	33.18	30.91	28.87	**41.93**	39.75	38.40	36.12	85.62	**86.11**	77.99	74.71	**0.204**	0.206	0.207	0.208
	RP4	**32.92**	32.52	30.44	28.47	**40.31**	39.12	37.76	35.52	82.23	**82.41**	75.16	72.41	**0.219**	0.221	0.221	0.222
	RP5	**31.76**	31.48	29.82	28.19	**38.86**	38.19	36.93	35.14	**76.70**	76.51	70.44	68.89	**0.229**	0.229	0.231	0.232
Painter	RP1	38.16	**40.41**	36.78	37.06	45.14	**47.50**	45.34	45.45	92.74	**94.67**	84.86	85.43	**0.267**	**0.267**	0.275	0.274
	RP2	37.44	**39.44**	36.48	36.69	44.30	**46.37**	44.63	44.65	91.15	**93.08**	84.15	84.94	**0.295**	**0.295**	0.302	0.301
	RP3	36.13	**37.64**	35.70	35.85	42.82	**44.51**	43.26	43.27	87.27	**88.90**	81.80	82.62	0.343	**0.342**	0.348	0.348
	RP4	35.35	**36.52**	35.08	31.80	41.98	**43.38**	42.36	39.19	84.07	**85.12**	79.48	80.15	0.378	**0.377**	0.382	0.382
	RP5	33.95	**34.74**	33.90	33.90	40.57	**41.62**	40.94	40.84	77.63	**78.38**	74.55	74.93	0.405	**0.403**	0.407	0.407
Frog	RP1	**31.79**	31.61	29.47	30.06	**41.17**	40.84	38.63	39.17	**92.58**	91.48	86.45	88.39	**0.067**	0.068	0.070	0.069
	RP2	**31.43**	31.25	29.27	29.82	**40.79**	40.46	38.37	38.87	**91.30**	90.09	85.41	87.28	**0.072**	**0.072**	0.075	0.074
	RP3	**30.87**	30.71	28.92	29.44	**40.13**	39.86	37.84	38.35	**88.94**	87.60	83.54	85.34	**0.079**	0.080	0.083	0.082
	RP4	**29.64**	29.49	28.12	28.50	**38.67**	38.39	36.71	37.13	**83.19**	81.72	78.91	80.29	**0.096**	0.099	0.099	0.099
	RP5	**27.88**	27.73	26.79	26.93	**36.43**	36.21	34.98	35.17	**73.97**	72.45	71.00	71.45	**0.119**	0.120	0.122	0.123
Carpark	RP1	**35.13**	35.03	33.55	33.73	43.17	**44.08**	43.19	43.66	**91.02**	90.77	89.58	89.40	0.173	**0.169**	0.171	0.171
	RP2	**34.98**	34.94	33.47	33.66	42.96	**43.88**	43.06	43.46	**90.74**	90.28	89.29	89.07	0.184	**0.180**	0.181	0.181
	RP3	**34.56**	**34.56**	32.99	33.22	42.34	**43.24**	42.32	42.80	**89.43**	89.02	87.65	87.54	0.196	**0.193**	0.194	0.194
	RP4	**34.06**	34.05	25.72	32.49	41.65	**42.54**	34.38	41.74	**87.82**	87.25	85.71	85.47	0.202	**0.197**	0.200	0.200
	RP5	**33.53**	33.37	25.68	31.90	41.07	**41.61**	34.27	40.96	**85.97**	85.11	83.50	83.03	0.214	**0.211**	0.213	0.215
Street	RP1	**37.95**	36.69	35.81	36.00	**46.53**	44.95	46.33	46.13	**91.80**	91.13	89.20	88.99	**0.120**	**0.120**	**0.120**	0.121
	RP2	**37.68**	36.43	35.59	35.77	**46.08**	44.45	45.94	45.74	**91.40**	90.57	88.69	88.44	**0.126**	**0.126**	0.127	0.127
	RP3	**37.46**	36.04	35.34	35.46	**45.67**	43.86	45.46	45.25	**90.98**	90.13	87.98	87.65	0.134	**0.133**	0.134	0.135
	RP4	**36.62**	35.35	34.61	34.62	**44.33**	42.85	44.13	43.88	**89.40**	88.42	86.07	85.35	0.150	**0.148**	0.151	0.152
	RP5	**35.42**	34.33	33.41	33.43	**42.34**	41.21	41.99	41.84	**86.28**	85.34	82.57	81.44	0.172	**0.171**	0.173	0.174
Hall	RP1	40.73	**41.01**	36.91	37.56	48.60	**49.60**	44.26	45.67	**92.26**	91.25	90.66	86.66	0.049	**0.048**	**0.048**	**0.048**
	RP2	38.79	**40.87**	36.96	37.54	45.72	**49.15**	44.32	45.59	**91.68**	91.22	90.50	86.55	0.056	**0.053**	0.054	0.055
	RP3	38.51	**40.71**	36.89	37.52	45.43	**48.78**	44.22	45.50	90.92	**91.04**	90.34	86.47	0.065	**0.064**	**0.064**	**0.064**
	RP4	38.48	**40.34**	36.85	37.23	45.22	**48.11**	44.15	44.95	**90.57**	90.22	90.03	86.16	0.077	**0.076**	**0.076**	**0.076**
	RP5	38.34	**39.52**	36.66	36.91	45.12	**46.73**	43.85	44.30	**89.92**	89.78	89.04	85.66	0.089	0.089	0.089	0.089
Mirror	RP1	35.34	**36.51**	34.05	33.26	41.57	**42.91**	40.20	39.45	90.24	**91.43**	87.44	85.88	**0.059**	0.060	0.063	0.066
	RP2	34.05	**35.42**	33.13	32.54	40.21	**41.82**	39.17	38.59	87.69	**88.95**	84.79	83.64	**0.074**	0.075	0.077	0.080
	RP3	32.76	**33.89**	31.96	31.55	38.85	**40.33**	37.79	37.50	83.56	**84.26**	80.84	79.55	**0.091**	**0.091**	0.093	0.096
	RP4	30.83	**31.48**	30.13	29.84	37.01	**38.02**	35.93	35.84	74.30	**74.66**	71.69	70.11	**0.116**	0.117	0.118	0.119
	RP5	28.49	**28.74**	27.79	27.55	34.36	**34.82**	33.38	33.23	**59.12**	58.73	56.45	54.63	**0.158**	0.161	0.160	0.162
Average	RP1	35.89	**36.17**	33.43	33.30	43.81	**44.06**	41.75	41.65	90.52	**90.80**	85.14	84.39	0.123	**0.122**	0.126	0.126
	RP2	35.20	**35.62**	33.19	33.07	42.93	**43.39**	41.40	41.29	89.49	**89.75**	84.29	83.64	0.136	**0.135**	0.139	0.139
	RP3	34.52	**34.86**	32.72	32.63	42.10	**42.54**	40.72	40.67	87.35	**87.52**	82.53	81.92	**0.152**	**0.152**	0.154	0.155
	RP4	33.53	**33.78**	31.14	31.39	40.89	**41.30**	38.83	39.20	**83.40**	83.24	78.92	78.14	**0.171**	0.172	0.174	0.174
	RP5	32.20	**32.25**	30.08	30.60	39.24	**39.43**	37.43	37.97	**76.47**	76.00	72.56	71.62	**0.194**	0.195	0.197	0.198

IVDE which was adapted to handle such compression-induced artefacts [7].

Even though the objective results show that synthesis results based on IVDE and DERS depth maps are better on average than synthesis results based on GANet and GWCNet, the LPIPS metric indicates that the average quality is very similar for all tested depth estimation methods. Most importantly, even though these methods were not fine-tuned on the MPEG test set, they present sufficient quality for indoors video sequences, and are competitive for outdoors sequences (Street and Carpark), as they are more similar to the KITTI driving images.

It is noticeable that the quality of depth maps degrades with the increase of baseline between a pair of views because GANet and GWCNet are also not optimized for such wide baseline stereo images. Again, this highlights a strong dependency towards the sequence properties that are similar to the ones of the training set, causing some lack of robustness for these methods. However, this indicates a very high potential of using Deep-Learning (DL) methods in the DSDE framework, as further improvements are possible by re-training the models on appropriate content, *i.e.,* considering multi-view high-resolution compressed textures as input, or optimization for view synthesis instead of depth fidelity.

TABLE XI

THE COMPARISON OF RUNTIMES FOR DECODING AND RENDERING

	Time for decoding and rendering of all views per one frame [s]			
	IVDE (CPU)	DERS (CPU)	GANet (GPU)	GWCNet (CPU)
Fan	494.27	978.75	78.99	337.58
Kitchen	442.56	992.29	124.00	397.42
Painter	669.04	1186.23	119.67	465.65
Frog	1266.24	1845.84	27.23	358.35
Carpark	385.32	489.35	62.65	264.33
Street	327.05	1033.02	100.98	311.67
Hall	224.21	597.89	29.18	136.99
Mirror	272.33	350.67	74.57	242.10
Average	510.13	934.26	77.16	314.26

For most of the sequences, in the case of synthesis using GWCNet depth maps, the compression with increasing QP yields in the synthesis quality which is closer to the synthesis quality obtained using IVDE and DERS depth maps. One may conclude that the quality of GWCNet depth maps does not depend so heavily on the quality of transported views, as in the case of classical methods. However, this behavior is also influenced by the limits of the proposed deep learning approaches for this type of content because the synthesis quality they can achieve for the high-quality and high-resolution transported views is somewhat saturated.

Subjectively both types of methods have their advantages and disadvantages. IVDE and DERS depth maps are noisier, but they have sharper object edges, while the two deep-learning-based methods produce depth maps that are somewhat cloudy and have smoother depth discontinuities. As a consequence, the synthesis results based on IVDE and DERS depth maps preserve the object edges better, whereas the results obtained using GANet and GWCNet depth maps often have ghosting artifacts around the objects (see a fragment of Kitchen sequence in Fig. 5). On the other hand, in some examples, the deep learning approaches better preserve the consistency of the objects which are uncovered (dis-occluded) temporally from one frame to another (see Carpark in Fig. 5).

Another aspect of performed experiments is the complexity of DSDE when different depth estimation methods are used. The estimation for different methods was performed using various computing hardware (of similar, yet not the same performance), nevertheless, to preserve the wholeness of presented results, these data are also provided in order to show the observed range of runtimes when using presented software, not for their direct comparison.

Table XI shows the overall runtime of decoding and rendering in MIV DSDE when different depth estimators are used for providing the depth maps. As it can be observed, the shortest time was observed when GANet is used, as it was the only method run on GPU. What should be underlined, the provided runtimes show the time required to estimate depth maps and render all views (so, *e.g.*, 25 views for Kitchen sequence), as it was required to fully evaluate the quality. Therefore, the decoding in MIV DSDE is much faster in practical real-world applications, as only one requested viewpoint has to be rendered at the time.

Moreover, the implementations of used depth estimators and MIV decoder were not optimized for the low computational complexity but are rather the implementations used for academic and standardization purposes. As for other conventional video codecs, the real-time implementations of each building block of the MIV decoder were already presented (for bitstream decoding [74] also for computationally expensive virtual view synthesis [74], [75]).

Fig. 5. Visual comparison of evaluated depth estimators for selected parts and viewports synthesized between positions of input views.

TABLE XII

BD-RATES SAVINGS AND RUNTIMES CHANGES OF ENCODING AND DECODING OF MIV DSDE (WITH GANET) OVER MIV ATLAS

Sequence	High-BR BD rate WS-PSNR	Low-BR BD rate WS-PSNR	High-BR BD rate IV-PSNR	Low-BR BD rate IV-PSNR	Atlas encoding	Video encoding	Decoding & Rendering
Fan	7.3%	-24.0%	22.5%	-15.0%	1.0%	80.6%	685.7%
Kitchen	---	---	---	---	0.6%	56.1%	837.9%
Painter	-62.0%	-69.3%	-67.3%	-70.4%	1.2%	74.8%	1110.8%
Frog	-2.2%	-28.5%	3.3%	-25.0%	1.2%	85.6%	553.7%
Carpark	102.4%	5.7%	-7.6%	-28.1%	1.7%	83.2%	637.8%
Hall	---	---	---	326.5%	1.6%	204.3%	634.0%
Street	261.8%	64.3%	-29.4%	-35.7%	2.0%	85.7%	633.1%
Mirror	8.0%	-32.4%	63.9%	-18.8%	1.0%	80.8%	900.2%

To conclude the results, Table XII shows the comparison of MIV Atlas (with IVDE-estimated depth maps) with MIV DSDE with GANet used for depth estimation. Even if the objective quality of rendered views was not the highest for this method, still, for most of the presented sequences it can be seen that for low bitrates the BD-Rate for IV-PSNR indicates gain in comparison of MIV Atlas. It further proves that MIV DSDE mode is a very efficient method for coding the immersive video, which is not highly dependent on the depth estimation method used at the decoder side.

V. Conclusion

This paper describes the motivation and technical details of the novel Decoder-Side Depth Estimation (DSDE) mode with Geometry Absent profile of the MPEG Immersive video (MIV) standard for efficient delivery of multiview immersive video.

By not transmitting the depth and shifting the depth estimation process to the decoder side, the DSDE brings the increment of compression efficiency and rendering quality when compared to MIV Encoder-Side Depth Estimation (ESDE) modes, especially for low bitrates. In addition, the usage of the Geometry Assistance SEI message, containing different features extracted from ground-truth or highly optimized depths at the encoder side further strengthens DSDE by allowing faster depth estimation together with accuracy improvement. The wide compatibility of DSDE to different depth estimators including the emerging Deep-Learning (DL) methods is investigated and the comparative results with the MIV main profile are derived.

Regarding future work, DSDE opens several research tracks. Any depth estimation method, even DL-based, is facing difficulties when handling specular regions, texture-less areas, and fine-geometry or complex objects. When high-quality depth information of such regions is available at the encoder side, efficiently delivering the relevant information in the format of video or SEI can be one of the improvement points of the DSDE mode. Moreover, even though DL-based depth estimation has achieved a significant performance gap with respect to the traditional methods, it is still immature in some DSDE aspects such as simultaneous support of diverse projection formats, arbitrary camera arrangements, and domain adaptation. Therefore, developing the DL-based techniques that support such versatility needs to be further studied.

The DSDE mode provides a significant reduction of encoder complexity (100 times faster than MIV encoder and 20% faster VVC encoding), nevertheless, from the perspective of the decoder, supporting DSDE can be fairly heavy in terms of processing time and memory capacity. Several optimized implementations already exist for decoding bitstreams and rendering the final synthesized virtual views. Still, the depth estimator which enables estimating precise depth maps with high spatial and depth resolutions in real-time is lacking for further deployment of DSDE in consumer devices. Fortunately, the presented scheme in some way future-proofs the MIV standard, as it is both agnostic to the video codec and to the depth estimator, so incoming innovations in these fields will provide even better quality for the final user.

References

[1] J. M. Boyce *et al.*, "MPEG immersive video coding standard," *Proc. IEEE*, vol. 109, no. 9, pp. 1521–1536, Sep. 2021.

[2] P. Garus, J. Jung, T. Maugey, and C. Guillemot, "Bypassing depth maps transmission for immersive video coding," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.

[3] D. Mieloch, O. Stankiewicz, and M. Domanski, "Depth map estimation for free-viewpoint television and virtual navigation," *IEEE Access*, vol. 8, pp. 5760–5776, 2020.

[4] A. Dziembowski, M. Domanski, A. Grzelka, D. Mieloch, J. Stankowski, and K. Wegner, "The influence of a lossy compression on the quality of estimated depth maps," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.

[5] L. Jorissen, P. Goorts, Y. Li, and G. Lafruit, *[FTV AhG] Soccer Light Field Interpolation Applied on Compressed Data*, document ISO/IEC JTC 1/SC 29/WG 11, MPEG2016/M37674, San Diego, CA, USA, Feb. 2016.

[6] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3268–3277.

[7] D. Mieloch, D. Klóska, and M. Woźniak, "Point-to-block matching in depth estimation," in *Proc. CSRN*, 2021, pp. 135–143.

[8] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4895–4904.

[9] *Manual of IVDE 3.0*, document ISO/IEC JTC1/SC29/WG4 MPEG2020/ N0058, Online, Jan. 2021.

[10] *Test Model 9 for MPEG Immersive Video*, document ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0084, Online, Apr. 2021.

[11] *Common Test Conditions for MPEG Immersive Video*, document ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0085, Online, Apr. 2021.

[12] *Text of ISO/IEC FDIS 23090-12 MPEG Immersive Video*, document ISO/IEC JTC1/SC29/WG4 MPEG2021/ N0111, Online, Jul. 2021.

[13] *Description of DERS*, document ISO/IEC JTC1/SC29/WG11 MPEG2020/ N19143, Brussels, Belgium, Jan. 2020.

[14] F. Zhang, V. Prisacariu, R. Yang, and P. H. S. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 185–194.

[15] P. Garus, F. Henry, J. Jung, T. Maugey, and C. Guillemot, "Immersive video coding: Should geometry information be transmitted as depth maps," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jul. 26, 2021, doi: 10.1109/TCSVT.2021.3100006.

[16] S. Rogge *et al.*, "MPEG-I depth estimation reference software," in *Proc. Int. Conf. 3D Immersion (IC3D)*, Dec. 2019, pp. 1–6.

[17] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[18] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and binocular stereo for depth estimation from images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 5, 2021, doi: 10.1109/TPAMI.2021.3070917.

[19] D. Mieloch, A. Dziembowski, and M. Domanski, "Depth map refinement for immersive video," *IEEE Access*, vol. 9, pp. 10778–10788, 2021.

[20] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1738–1764, Apr. 2022.

[21] L. Fang, Y. Xiang, N.-M. Cheung, and F. Wu, "Estimation of virtual view synthesis distortion toward virtual view position," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1961–1976, May 2016.

[22] Y. Zhang, W. Dai, M. Xu, J. Zou, X. Zhang, and H. Xiong, "Depth estimation from light field using graph-based structure-aware analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4269–4283, Nov. 2020.

[23] A. Chuchvara, A. Barsi, and A. Gotchev, "Fast and accurate depth estimation from sparse light fields," *IEEE Trans. Image Process.*, vol. 29, pp. 2492–2506, 2020.

[24] Y. Li, Q. Wang, L. Zhang, and G. Lafruit, "A lightweight depth estimation network for wide-baseline light fields," *IEEE Trans. Image Process.*, vol. 30, pp. 2288–2300, 2021.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[27] *Depth Map Formats Used Within MPEG 3D Technologies*, document ISO/IEC JTC1/SC29/WG11 MPEG2017/N16730, Geneva, Switzerland, Jan. 2017.

[28] M. Milovanovic, F. Henry, M. Cagnazzo, and J. Jung, "Patch decoder-side depth estimation in mpeg immersive video," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1945–1949.

[29] C. Won, J. Ryu, and J. Lim, "OmniMVS: End-to-end learning for omnidirectional stereo matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8986–8995.

[30] C. Won, J. Ryu, and J. Lim, "SweepNet: Wide-baseline omnidirectional depth estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6073–6079.

[31] N.-H. Wang, B. Solarte, Y.-H. Tsai, W.-C. Chiu, and M. Sun, "360SD-Net: 360° stereo depth estimation with learnable cost volume," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 582–588.

[32] *Software manual of IV-PSNR for Immersive Video*, document ISO/IEC JTC1/SC29/WG04 MPEG2020/ N0013, Online, Oct. 2020.

[33] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD986 Curves*, document ISO/IEC JTC 1/SC 29/WG 11, MPEG2001/M15378, 2001.

[34] B. Kroon, *3DoF+ Test Sequence ClassroomVideo*, document ISO/IEC JTC1/SC29/WG11 MPEG2018/M42415, San Diego, CA, USA, Apr. 2018.

[35] L. Ilola, V. K. M. Vadakital, K. Roimela, and J. Keränen, *New Test Content for Immersive Video Nokia Chess*, document ISO/IEC JTC1/SC29/WG11 MPEG2019/M50787, Geneva, Switzerland, Sep. 2020.

[36] R. Doré, *Technicolor 3DoF+ Test Materials*, document ISO/IEC JTC1/SC29/WG11 MPEG2018/M42349, San Diego, CA, USA, Mar. 2018.

[37] R. Doré, G. Briand, and F. Thudor, *InterdigitalFan Content Proposal for MIV*, document ISO/IEC JTC1/SC29/WG11 MPEG/M54732, Jun. 2020.

[38] R. Doré, G. Briand, and F. Thudor, *InterdigitalGroup Content Proposal*, document ISO/IEC JTC1/SC29/WG11 MPEG2020/M54731, Jun. 2020.

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

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

[41] D. Mieloch, A. Dziembowski, and M. Domański, *Natural Outdoor Test Sequences*, document ISO/IEC JTC1/SC29/WG11 MPEG2019/M51598, Brussels, Belgium, Jan. 2020.

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

[43] B. Salahieh *et al.*, *Kermit Test Sequence for Windowed 6DoF Activities*, document ISO/IEC JTC1/SC29/WG11 MPEG2018/M43748, Jul. 2018.

[44] D. Mieloch, A. Dziembowski, and M. Domański, *Natural Outdoor Test Sequences*, document ISO/IEC JTC1/SC29/WG11 MPEG2020/M51598, Brussels, Belgium, Jan. 2020.

[45] D. Doyen *et al.*, *Light Field Content From 16-Camera Rig*, document ISO/IEC JTC1/SC29/WG11 MPEG2017/M40010, Geneva, Switzerland, Jan. 2017.

[46] Y. Yao, "MVSNet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 767–783.

[47] D. Shin, "How do users experience the interaction with an immersive screen," *Comput. Hum. Behav.*, vol. 98, pp. 302–310, Sep. 2019.

[48] G. Esakki, A. S. Panayides, S. Teeparthi, and M. S. Pattichis, "A comparative performance evaluation of VP9, x265, SVT-AV1, VVC codecs leveraging the VMAF perceptual quality metric," *Proc. SPIE*, vol. 11510, pp. 181–190, Aug. 2020.

[49] Y. Chan, C. Fu, H. Chen, and S. Tsang, "Overview of current development in depth map coding of 3D video and its future," *IET Signal Process.*, vol. 14, no. 1, pp. 1–14, Feb. 2020.

[50] A. Wieckowski *et al.*, "VVenC: An open and optimized VVC encoder implementation," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMEW)*, Jul. 2021, pp. 1–2.

[51] Y. Sun, A. Lu, and L. Yu, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1408–1412, Sep. 2017.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[53] *Toward a Practical Perceptual Video Quality Metric*. Accessed: Feb. 29, 2022. [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[54] K. Han, W. Xiang, E. Wang, and T. Huang, "A novel occlusion-aware vote cost for light field depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 18, 2021, doi: 10.1109/TPAMI.2021.3105523.

[55] B. Huang, H. Yi, C. Huang, Y. He, J. Liu, and X. Liu, "M3 VSNET: Unsupervised multi-metric multi-view stereo network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3163–3167.

[56] *Text of ISO/IEC DIS 23090-5 Visual Volumetric Video-based Coding and Video-based Point Cloud Compression 2nd Edition*, document ISO/IEC JTC1/SC29/WG7 MPEG2021/N00188, Jul. 2021.

[57] *2021 TV Video Specifications*. Accessed: Jan. 11, 2021. [Online]. Available: https://developer.samsung.com/smarttv/develop/specifications/media-specifications/2021-tv-video-specifications.html

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

[59] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[60] A. Dziembowski, D. Mieloch, O. Stankiewicz, M. Domanski, G. Lee, and J. Seo, "Virtual view synthesis for 3DoF+ video," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.

[61] D. Bonatto, S. Fachada, S. Rogge, A. Munteanu, and G. Lafruit, "Real-time depth video-based rendering for 6-DoF HMD navigation and light field displays," *IEEE Access*, vol. 9, pp. 146868–146887, 2021.

[62] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3827–3837.

[63] M. K. Yucel, V. Dimaridou, A. Drosou, and A. Saa-Garriga, "Real-time monocular depth estimation with sparse supervision on mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2428–2437.

[64] H.-N. Hu *et al.*, "Joint monocular 3D vehicle detection and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5389–5398.

[65] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methods—A state-of-the-art review," *Sensors*, vol. 20, no. 8, p. 2272, Apr. 2020.

[66] J. Yang, Z. Jiang, and X. Ye, "Depth super-resolution with color guidance: A review," in *RGB-D Image Analysis and Processing*. Cham, Switzerland: Springer, 2019, pp. 51–65.

[67] M. M. Ibrahim, Q. Liu, R. Khan, J. Yang, E. Adeli, and Y. Yang, "Depth map artefacts reduction: A review," *IET Image Process.*, vol. 14, no. 12, pp. 2630–2644, Oct. 2020.

[68] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German Conf. Pattern Recognit. (GCPR 2014)*, pp. 31–42.

[69] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, "Continuous 3D label stereo matching using local expansion moves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2725–2739, Nov. 2017.

[70] H. Xu, X. Chen, H. Liang, S. Ren, Y. Wang, and H. Cai, "CrossPatch-based rolling label expansion for dense stereo matching," *IEEE Access*, vol. 8, pp. 63470–63481, 2020.

[71] P. Ji, J. Li, H. Li, and X. Liu, "Superpixel alpha-expansion and normal adjustment for stereo matching," *J. Vis. Commun. Image Represent.*, vol. 79, Aug. 2021, Art. no. 103238.

[72] J. Seuffert, A. Grassi, T. Scheck, and G. Hirtz, "A study on the influence of omnidirectional distortion on CNN-based stereo vision," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 809–816.

[73] M. Poyser, A. Atapour-Abarghouei, and T. P. Breckon, "On the impact of lossy image and video compression on the performance of deep convolutional neural network architectures," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2830–2837.

[74] M. Chen, B. Salahieh, M. Dmitrichenko, and J. Boyce, "Simplified carriage of MPEG immersive video in HEVC bitstream," *Proc. SPIE*, vol. 11842, Aug. 2021, Art. no. 118420C.

[75] J. Fleureau, B. Chupeau, F. Thudor, G. Briand, T. Tapie, and R. Dore, "An immersive video experience with real-time view synthesis leveraging the upcoming MIV distribution standard," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMEW)*, Jul. 2020, pp. 1–2.

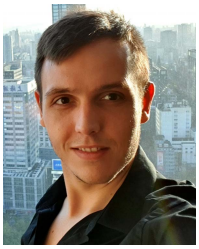
[76] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, "Standardization status of immersive video coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 5–17, Mar. 2019.

[77] M. Domanski, Y. Al-Obaidi, and T. Grajek, "Universal modeling of monoscopic and multiview video codecs with applications to encoder control," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 2144–2148, doi: 10.1109/ICIP42928.2021.9506735.

[78] G. Clare *et al.*, *[MIV] Combination of m56626 and m56335 for Geometry Assistance SEI Message*, document ISO/IEC JTC1/SC29/WG4 MPEG2021/ m56950, Online, Apr. 2021.

[79] Y. Liu, J. Liu, A. Argyriou, L. Wang, and Z. Xu, "Rendering-aware VR video caching over multi-cell MEC networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2728–2742, Mar. 2021.
[80] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.
[81] *Summary of the Results of the Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation*, document ISO/IEC JTC 1/SC 29/WG 11, MPEG2016/N16318, Geneva, Switzerland, Jun. 2016.
[82] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.
[83] P. Gao, W. Xiang, and D. Liang, "Texture-Distortion-Constrained joint source-channel coding of multi-view video plus depth-based 3D video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3326–3340, Nov. 2019.
[84] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Fast 3D-HEVC depth map encoding using machine learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 850–861, Mar. 2020.
[85] M. Cheon and J.-S. Lee, "Subjective and objective quality assessment of compressed 4k UHD videos for immersive experience," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 7, pp. 1467–1480, Jul. 2018.
[86] *MPEG Immersive Video (MIV) Content Database*. Accessed: Feb. 25, 2022. [Online]. Available: https://mpeg-miv.org/index.php/content-database-2/

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



Patrick Garus received the M.Sc. and M.Ed. degrees from Rheinisch-Westfälische Technische Hochschule Aachen (RWTH Aachen), Aachen, Germany, in 2018 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Orange Labs and INRIA, Rennes, France. He worked as a Research Assistant with the Institute of Communication Engineering (IENT), Aachen, from 2015 to 2017, on synthesis of dynamic textures for video coding. His current research interests include video compression, immersive video coding, and related machine learning-based technologies for standardization activities in MPEG.

Marta Milovanović received the B.Sc. degree in electrical and computer engineering, focused on signal processing, from the School of Electrical Engineering, University of Belgrade, Serbia, in 2018, and the M.Sc. degree in multimedia networking from the Télécom Paris, University of Paris-Saclay, France, in 2019, where she is currently pursuing the Ph.D. degree with the Institut Polytechnique de Paris. She is also a Research Engineer with the Orange Labs, France. Her current research interest includes immersive video coding and processing.

Joël Jung (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Nice Sophia Antipolis, France, in 2000, and the habilitation degree in electrical engineering from Sorbonne University, Paris, France, in 2019. From 1996 to 2000, he was with the CNRS Laboratory, Sophia Antipolis, active in the improvement of video decoders based on the correction of compression and transmission artifacts. In 2000, he joined Philips Research, Paris, as a Research Scientist in video coding, postprocessing, perceptual models, objective quality metrics, and low-power codecs. He worked with the Orange Labs, France, from 2004 to 2020. He has contributed to the 2D and 3D video coding standard HEVC/3D HEVC. He is currently the Principal Researcher with the Tencent Media Laboratory, Palo Alto, USA. He is involved in the standardization of immersive video codecs, as the Co-Chair of the MPEG Immersive Video (MIV) Group, dealing with coding, view synthesis, and depth estimation with six degrees of freedom. His current research interests include video quality evaluation for gaming content and immersive video content, contributing to the ITU-T Study Group 12 (performance, QoS, and QoE) and being the Chair of the Immersive Video Focus Group of MPEG Advisory Group 5 (AG5) on video quality assessment.

Jun Young Jeong received the B.S. and M.S. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2013 and 2016, respectively. He has been a Research Staff with the Immersive Media Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, since 2016, and has been primarily 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.

Smitha Lingadahalli Ravi received the bachelor's degree in engineering from Visvesvaraya Technological University, India, in 2018. She is currently pursuing the joint Ph.D. degree with the Orange Labs, Cesson-Sévigné, and the L'Institut National des Sciences Appliquées de Rennes. Her current research interests include depth estimation and view synthesis for immersive video content.

Basel Salahieh (Senior Member, IEEE) received the B.S. degree in communication engineering from Aleppo University, Syria, in 2007, the M.S. degree in electrical engineering from The University of Oklahoma, OK, USA, in 2010, and the M.S. degree in optical science and the Ph.D. degree in electrical and computer engineering from The University of Arizona, AZ, USA, in 2015. He is the Founder and the Chief Technology Officer of Vimmerse Inc., CA, USA, responsible for delivering immersive media solutions for businesses and end users. Prior to that, he worked as a Immersive Media Standards Architect at Intel Corporation. His research interests are related to light fields, point clouds and meshes, extended reality, and immersive video systems. He also serves as an Editor for the *Test Model of MPEG Immersive Video*.