The MPEG Immersive Video Standard – Current Status and Future Outlook

Vinod Kumar Malamal Vadakital, Ofinno LLC, Tampere, Finland

Adrian Dziembowski, Poznań University of Technology, Poznań, Poland

Gauthier Lafruit, Université Libre de Bruxelles, Brussels, Belgium

Franck Thudor, InterDigital, France

Gwangsoon Lee, Electronics and Telecommunications Research Institute, Daejeon, S. Korea

Patrice Rondao Alface, Nokia Technologies, Antwerp, Belgium

***Abstract*—**The MPEG immersive video (MIV) standard is the latest addition to the MPEG-I suite of standards. It focuses on the representation and coding of immersive media. MIV is designed to support virtual and extended reality (VR/XR) applications that require six degrees of freedom (6DoF) visual interaction with the rendered scene. Edition-1 of MIV is now in its final phase of standardization. Leveraging conventional 2D video codecs, the MIV standard efficiently codes volumetric scenes and allows advanced visual effects like bullet-time fly-throughs. The video feeds capturing the scene are first processed to identify a set of basic views that are augmented with additional information from all other views. The data is then intelligently packed into atlases and further compressed with any existing 2D video codec of choice. Experimental results show BD-PSNR gains of up to 6 dB in the 10 to 20 Mbps range compared to a naive simulcast multiview video coding approach. The paper concludes with an outlook on future extensions for the second edition of MIV.

The Moving Picture Experts Group (MPEG) has been developing audio-visual coding standards for over three decades. Its main goal is to standardize audio-visual coding technologies that enable efficient storage and interchange formats. For instance, the MPEG-2 [1] video coding standard was the first to serve the digital television era. The Advanced Video Coding standard (AVC) [2] and the High-Efficiency Video Coding (HEVC) standard [3] followed MPEG-2 under the MPEG-4 umbrella. Today, the video standards from MPEG cater to a wide variety of heterogeneous digital devices. These devices range from webcams and smartphones to camcorders and television set-top boxes. More recently, MPEG completed the specification of the Versatile Video Coding (VVC) [4]. VVC compresses video more efficiently than AVC and HEVC [5].

Improving the coding efficiency of 2D video is still a hot topic in MPEG. Nevertheless, some years back, MPEG also started to focus on the compression of 3D immersive audio-visual content covered by the MPEG-I suite of standards; the suffix "I" in MPEG-I signifies immersion. The MPEG immersive video (MIV) standard [6] is part of the MPEG-I family of standards.

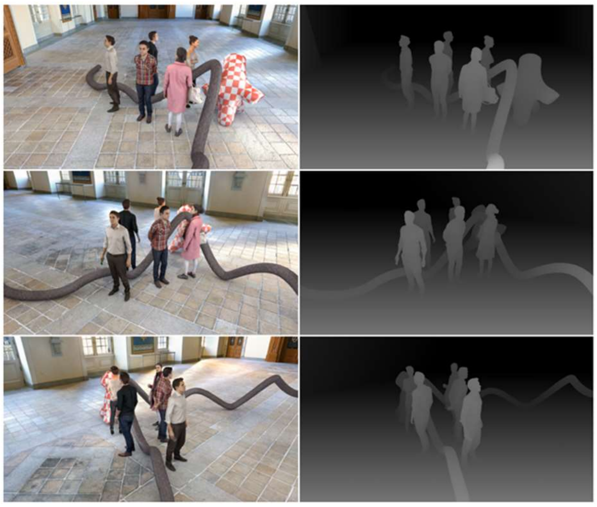


Figure 1: A frame of a multiview plus depth format from 3 cameras captured at the same time instant.

MIV efficiently codes a multiview plus depth (MVD) video representation of a 3D scene, where a sparse set of cameras, each having an arbitrary pose, capture information about the 3D scene. Figure 1 shows an example of an MVD frame from three different cameras.

The most straightforward approach to compressing a multiview representation is called simulcast coding which compresses each view independently. However, simulcasting does not consider inter-view redundancies and can incur a high bitrate penalty. MIV, unlike simulcasting, considers inter-view redundancies during coding and provides better compression by also exploiting geometry information using depth maps.

Recovering the depth maps from a decoded MIV bitstream offers a viewer six degrees of freedom (6DoF) to render the decoded scene. A 6DoF representation - unlike three degrees of freedom (3DoF) - provides a larger viewing space, where viewers have both translational and rotational freedom of movement at their disposal. In fact, the absence of motion-parallax in 3DoF videos is inconsistent with the human visual perception and often leads to visual discomfort. This is resolved using 6DoF, where the visual perspective of the scene coherently changes with the viewer's pose thanks to a renderer that synthesizes the required perspective video views at a high level of realism [7, 8, 9, 10].

FROM 360 VIDEO TO 6DOF

A 360-degree omnidirectional video is an example of a 3DoF representation. A viewport from the 360-degree video is selected by changing the head pose. However, in a 360-degree video, the rendered viewport is responsive only to rotational motion. A left, right, forward, or backward head movement does not result in a corresponding translation movement of the viewport. This discrepancy creates an awkward visual effect where objects in the scene follow the viewer, often resulting in VR sickness.

In contrast, the MIV standard primarily targets Virtual and Extended Reality (VR/XR) 6DoF [11] use cases. One example of a commercially relevant VR use case is sports broadcasting. The viewer can watch a sports event from any desired position within a viewing volume or visualize the scene using visual effects such as the bullet-time fly-throughs. Other practical and commercial use cases for MIV include telepresence, immersive training videos, and virtual tourism.

MIV extends the Visual Volumetric Video-based Coding (V3C) bitstream format specified in [12]. While MIV targets use cases for visualizing any arbitrary viewpoint to the scene without any tactile interaction, other MPEG-I tools support collision detection with 3D geometric shapes for more advanced AR/XR applications. MPEG produced the Video-based Point Cloud Coding (V-PCC) standard [12] for this purpose, while currently studying extensions for dynamic mesh coding. They are all part of the more generic V3C standard specification supporting a plethora of AR/XR use cases.

This article primarily focuses on the MIV-related V3C aspects. The next section provides more details about the input formats supported by MIV edition-1.

SCENE INPUT FORMATS

In MIV, the processing of input video frames produces smaller images, called patches, which are packed into mosaics, called atlases. Edition-1 of the MIV standard supports two types of input formats. The first is a multiview texture (plus depth) format, and the second is a multiplane/multisphere input format.

Multiview plus depth (MVD) input format

The MVD input format is a set of videos, called source views, captured by a group of cameras having an arbitrary pose. The videos from each source view represent a projection of a part of a volumetric scene onto the camera projection plane. Each video referenced from a source view describes either projected geometry (depth with an optional occupancy map) or attributes. These attributes typically include texture. However, MIV also supports other attributes such as surface normals, material maps, reflectance, and transparency.

Additional metadata provided for each source view include:

- the bit depth of the source videos for both geometry and attributes,
- camera intrinsic data like the focal length and principal point,
- projection-plane dimensions,
- camera extrinsic data like the camera pose, and
- the camera projection format

The MIV standard supports perspective, equirectangular (ERP), and orthographic projection formats.

Multiplane and multisphere (MPI/MSI) input format

The multiplane image format (MPI) is a layered representation of a 3D scene viewed from a reference view. The location of the reference view is at the centre of the camera rig. Constructing an MPI from an MVD is done by un-projecting the pixels from the original cameras into 3D space and re-projecting them back onto the layers of a chosen reference camera. For a perspective reference camera, the layers are fronto-parallel planes, as shown in the left column of Figure 2. In the case of an ERP reference camera, the layered representation takes the form of a multisphere image (MSI), where sampling at different radii generates the layers. The right column of Figure 2 shows an illustration of an MSI representation. The generation of the MPI/MSI is outside the scope of the MIV standard.

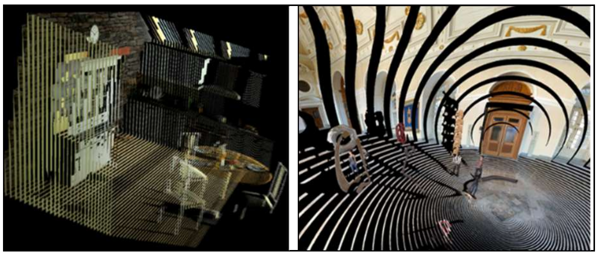


Figure 2: Multiplane and multisphere image representation.

Each depth layer of an MPI/MSI video frame is processed into texture patches with constant depth and then packed into atlases by the encoder. The encoder architecture and the process of atlas generation are detailed next.

MIV ENCODER ARCHITECTURE

Rather than compressing each captured view separately, MIV compresses all source views into atlases that contain patches. A typical MIV encoder selects a sub-set of source views with minimum redundant information between them. These views are called basic views. From the remaining source views, information that is not available in the basic views is collected as patches. The patches are packed together into mosaic-like images called atlases, an example of which is shown in Figure 4.

These patches are identified by un- and re-projecting (back and forth between 2D and 3D space) the pixels of a source view onto another source view using their depth information. By doing so, invisible regions from one view may become visible in another view as disocclusions. Disocclusions generally occur at object boundaries and therefore often create irregularly-shaped patches that are then packed into one or more attribute (texture) and geometry (depth) atlases.

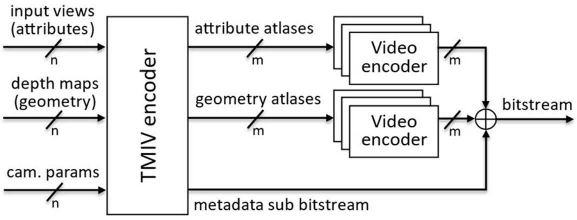


Figure 3: A high-level block diagram of a typical MIV encoder.

The dimensions of patches are optimized to reduce inter-view redundancy and minimize the number of pixels a decoder and renderer should process to generate a viewport. 2D video encoders encode the resulting geometry and attribute atlases. Metadata that describes the atlases is encoded using the MIV standard. Figure 3 shows a block diagram of a typical MIV encoder, and Figure 4 illustrates the concept of patches and atlases.

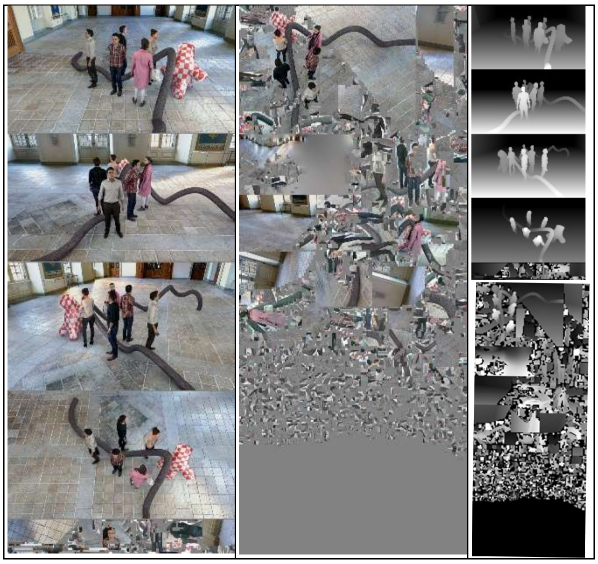


Figure 4: An example illustration of atlases: two attribute atlases (left and middle columns) and corresponding two geometry atlases (right column, top and bottom) with reduced resolution.

The following sub-sections discuss the main stages of the MIV encoding pipeline, including the selection of basic views, pruning, packing, post-processing of atlases, and the MIV bitstream generation.

Basic views selection

As the first step towards encoding of a 3D scene, a MIV encoder chooses a subset of views, called basic views, from the set of source views. Inter-view redundancies between views chosen as basic views are minimal. No pruning operation is performed on the basic views and they are packed into atlases as complete views. The rest of the input views, called additional views, are either pruned and packed as a mosaic of small patches or omitted entirely, depending on the chosen profile, cf. section [PROFILES].

The basic views are automatically selected based on the camera arrangement, using the partitioning around medoids (PAM) algorithm [13]. The number of basic views is configurable at the encoder. If the configuration requires only one basic view, the view captured by the most central camera in the camera rig is chosen as the basic view. If the configuration requires *k* basic views (and *k* > 1), the *k* views that are most distant from each other are selected.

Pruning of additional views

The pruning process minimizes inter-view redundancy between source views and determines if a pixel in an additional view should be removed or preserved for encoding. The output from a pruner is a hierarchical graph of views called the pruning graph. The set of basic views, established from the view selection process in the previous step, is at the top of the view hierarchy. They form the root node of the pruning graph (N_0 in Figure 5). Pixels from the basic views are projected onto each additional view. Each pixel of the additional view is then classified to be pruned (discarded) if it has similar geometry and luminance as the pixel in the basic view. Otherwise, the pixel is preserved (unpruned).

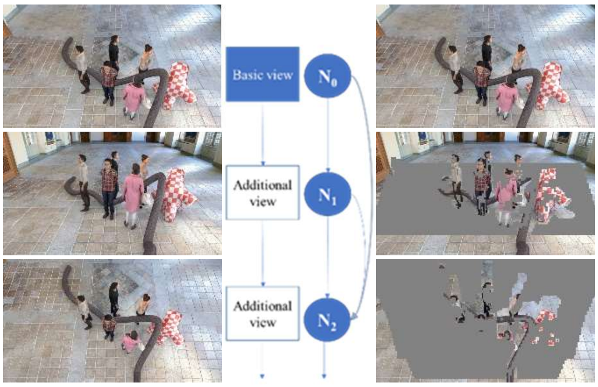


Figure 5: Pruning process; left: source views, middle: pruning graph, right: pruned views (grey regions denote pruned pixels).

Subsequently, the pixels of every additional view, higher up in the pruning graph, are projected onto the remaining additional views, as illustrated in Figure 5. This pixel classification process is repeated until all pixels in all additional views are classified to be pruned or preserved.

t

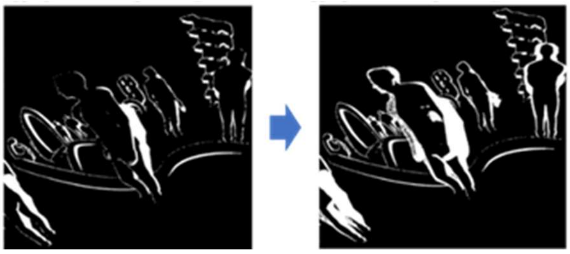


Figure 6: Aggregation of the pruning mask, showing the increase of the unpruned pixels in the moving region; left: frame i, right: frame i + k.

The pruning masks should be made as coherent as possible across adjacent atlas video frames for encoding the atlas videos efficiently. Therefore, the pruning masks are accumulated over a specified number of consecutive input source video frames, which increases the number of unpruned pixels, especially in regions with motion. Figure 6 illustrates the accumulation of the pruning masks.

Packing into atlases

After pruning, the views may contain both pruned and unpruned regions. To further improve coding efficiency, unpruned pixel regions, called patches, from the *n* input views are gathered and packed into *m* atlases, where *m* is usually much smaller than *n*.

The reference software for MIV uses the MaxRect algorithm [14] for packing the patches efficiently. First, all patches are sorted in decreasing order of their dimensions. Then, each patch is inserted into an atlas using the MaxRect algorithm. A MIV bitstream signals the original spatial position of each patch, its size and the view index from which the patch is extracted.

Atlas processing and bitstream formation

After packing patches into atlases, the atlases are further processed by some optional image filtering operations to improve video compression performances. Both attribute (texture) and geometry atlases are post-processed.

An attribute atlas is post-processed by modifying the average colour of each patch to reduce the number and intensity of edges between patches and unoccupied atlas regions. A geometry atlas is post-processed in two ways; first by modifying its dynamic range and second by decreasing its spatial resolution. [15] provides a detailed description of processing performed on the texture attribute atlas. [15] and [16] describe coding and downscaling of the geometry atlases.

Finally, each attribute and geometry atlas is separately encoded using a regular 2D video encoder, e.g., using the most advanced video codec to date (still under development/finetuning), the Versatile Video Codec (VVC), or a stable open implementation thereof, the VVenC [17] implementation, that we have used for the MIV evaluations in this article. As a matter of fact, MIV is video codec agnostic and rather focuses on transforming 3D scene information (or its 2D projections) into an atlas representation that can easily be handled by any 2D video codec; it's up to the MIV codec developer to decide which 2D video codec to use inside.

In the final aggregation step, all video sub-bitstreams combined with their associated metadata are multiplexed into a single decodable MIV-compliant bitstream [16] suitable for storage or transmission.

The following section provides a brief overview of the architecture of a MIV decoder.

MIV DECODER ARCHITECTURE

At the decoder, the multiplexed MIV bitstream is demultiplexed into a metadata sub-bitstream and video sub-bitstreams for all attribute and geometry atlases. Each video sub-bitstream is decoded using independent 2D video decoder instantiations, as shown in Figure 7.

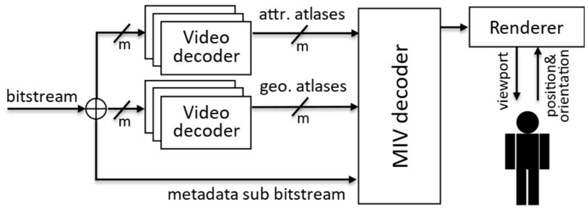


Figure 7: A high-level block diagram of a typical MIV decoder

Reconstructing the (pruned) source views is done by unpacking the (pruned) source views from the decoded video atlases. The reconstructed source views, together with their corresponding depth maps, are used to render the viewport requested by a viewer. If no geometry is transmitted, e.g., when using the MIV Geometry Absent profile (cf. PROFILES section), an additional depth estimation step is done before the rendering [18].

Different applications require different tool sets and facilities from the MIV specification for their operation. The MIV standard gathers tools suitable for applications into groupings called profiles. A listing of profiles specified in edition-1 of MIV, along with a brief description of their use, follows.

PROFILES SUPPORTED BY MIV

Like most MPEG codecs, MIV caters to different use cases by means of profiles. Each profile is a collection of features that is normatively enabled by the specification to target different application classes. A comprehensive description of these profiles and their supported tool-sets can be found in Annex-A of [6]. Edition-1 of MIV supports the following three profiles.

- **MIV Main Profile**

This profile provides the basic facilities that are required by VR applications. It is suitable for applications that use MVD videos as input. This profile does not support a separate occupancy map, and geometry and attribute atlases are coded as independent videos.

- **MIV Extended Profile**

This profile extends the facilities provided by the MIV main profile with additional tools such as the support for external occupancy maps. Occupancy maps are additional videos containing binary data that indicates if a co-located pixel in the related geometry and texture videos belongs to a valid 3D point in the scene. This profile also allows geometry and attribute data to be packed together, rather than separately as supported by the MIV Main profile. This facility is found to be useful in reducing the number of decoder instances at the client.

The profile also includes a sub-profile, called the Restricted Geometry Profile, for applications that use MPI/MSI videos as inputs. In this sub-profile, only texture and transparency attributes are coded as video sub-bitstreams. The MPI/MSI MIV encoding is suitable for real-time rendering in low-end devices because the rendering algorithm is computationally less complex.

- **Geometry Absent Profile**

This profile is suitable for applications with computationally powerful decoders that can perform real-time depth estimation. It may also be used to capture multiview data without depth for further depth estimation in the cloud (not necessarily real-time). This profile encodes only the texture attribute data in the bitstream. Geometry is estimated using a client-side depth estimator, referred to as the Decoder Side Depth Estimation (DSDE), in the remainder of this article.

The next section describes the test model created to evaluate the coding and synthesis performance of algorithms used by MIV for some selected profiles.

EVALUATION OF THE EDITION-1 TEST MODEL

During the development of the MIV standard, many tools and improvements were proposed over time to improve coding and synthesis performance. In order to evaluate and compare each proposal, the MIV Common Test Conditions (MIV-CTC) [20] were defined, allowing multiple organizations to evaluate their proposals in the exact same way. The tools considered promising are put in the so-called Test Model software suite for further evaluation [16]. Eventually, a subset of the tools is retained as the standard reference software, alongside of the standard description document published worldwide.

The viewport generation of a MIV scene by a renderer is beyond the scope of the MIV standard. Individual vendors can implement their own renderers to synthesize novel viewports from the coded scene. For the test model, various 6DoF view synthesizers [8, 10, 19] were explored during the MIV standardization activities. Further details on the rendering process can be found in [16].

The remaining part of this section briefly describes the test conditions and summarizes the experimental results obtained during the evaluation of the test model.

Common Test Conditions

Apart from contextual test parameters, the MIV-CTC specifies test sequences, the entire encoding and decoding pipeline (including the exact version and configuration for the used software), and the methodology for assessing the coding efficiency.

Test sequences

The test set defined in the MIV-CTC comprises 16 test sequences, including natural and computer-generated (CG) content, captured by perspective and omnidirectional (ERP) cameras. The sequences differ in resolution (from FullHD to 4K), the number of views (9 to 25), and camera arrangement (including simple, linear camera arrangements, camera arrays, and systems with cameras placed on an arc or sphere). The MIV-CTC [20] provides a detailed description of all the test sequences.

Video coding and quality assessment

As mentioned before, the MIV standard is codec-agnostic. Therefore, video encoding can use codecs like HEVC or VVC. MIV-CTC uses the VVenC VVC implementation.

The CTC evaluates the video coding performance at five rate points (five different bitrates). These rate points are encoded using appropriately chosen quantization parameters (QP) values. The five rate points are independently selected for each test sequence to obtain valid rates and meaningful rate-distortion curves. Besides bitrate constraints, the tests limit the number of pixels decoded per frame. The MIV-CTC uses the same limit defined for HEVC level 5.2. The HEVC level 5.2 allows as many luma samples as an 8K video with a frame rate of 30 frames per second.

Furthermore, the MIV-CTC defines the methodology to assess objective and subjective quality. Objective quality is evaluated using two full-reference quality metrics: WS-PSNR [21] and IV-PSNR [22]. These metrics measure the quality of synthesized source views by calculating BD rates [23].

Subjective evaluation of quality uses videos generated using pose traces. Pose traces are predefined camera paths traversing the scene's viewing volume and can differ between sequences. Pose traces mimic the virtual navigation of a viewer. It also ensures that all subjective test participants watch and evaluate the same video. A statistical Mean Observation Score (MOS) is gathered across all pose traces and at the different rate points to decide which software tools in the MIV Test Model provide the most satisfying visual experience.

Experimental results

This section presents the evaluation of coding efficiency of MIV with experimental results, using two profiles of MIV: Main and Geometry Absent (GA). The experiments were conducted by following MIV-CTC conditions [20] and used the Test Model for MPEG Immersive video 11.0 (TMIV 11.0) [15].

The results of MIV Main and MIV GA are compared against the multiview simulcast approach, where several full views and depth maps are independently encoded using VVenC.

The experiments used the same renderer in the three tested cases to keep comparisons fair. Following MIV-CTC conditions, all tests used the same pixel rate. Due to this pixel rate constraint, the number of coded views in MIV GA and multiview simulcast approaches had to be lowered, which resulted in visual artefacts not present when encoding sequences using the MIV Main profile. The lack of artefacts when using the MIV Main profile is because it judiciously uses basic and pruned additional views, well-packed into small atlases.

Table 1 provides results of multiview simulcast compared against MIV Main and MIV GA profiles. The table lists percentage BD-rate values. Negative values indicate that MIV reduces the total bitrate of the video while preserving the same quality. If the difference between two tested approaches cannot be reliably estimated, the gain or loss is highlighted only by the colour of the table cell (i.e., green denotes gains and red losses).

Table 1: BD-rates of Multiview simulcast vs. MIV Main and MIV Geometry Absent (negative number indicates better efficiency of MIV).

Type		Sequence	Multiview simulcast vs. MIV Main		Multiview simulcast vs. MIV GA	
			Y-PSNR	IV-PSNR	Y-PSNR	IV-PSNR
Computer-generated (CG)	ERP	Chess	---	---	681.4%	-32.6%
		ChessPieces	---	---	---	-8.6%
		ClassroomVideo	-24.6%	-16.2%	129.4%	25.8%
		Hijack	-54.1%	-59.5%	---	---
		Museum	-18.8%	-25.8%	133.6%	42.3%
	Perspective	Cadillac	-4.3%	-24.8%	-74.6%	-72.6%
		Fan	-32.5%	-44.4%	-91.8%	-83.7%
		Kitchen	-34.6%	-54.9%	-39.0%	-24.9%
		Mirror	-38.5%	-45.9%	-67.9%	-67.6%
Natural content		Carpark	-47.6%	-50.5%	-64.0%	-61.9%
		Fencing	-32.2%	-33.1%	-38.8%	-54.3%
		Frog	-6.1%	-24.4%	-61.5%	-61.7%
		Hall	-77.6%	-68.3%	-88.2%	-63.7%
		Painter	-16.2%	-29.3%	-68.2%	-58.8%
		Street	-18.8%	-40.7%	-51.1%	-56.9%

As presented in the left column of Table 1, MIV Main allows encoding the multiview sequence much more efficiently than the multiview simulcast approach, significantly reducing bitrate, especially for the omnidirectional content. For these sequences, only a small subset of source views is coded in the multiview simulcast approach. In this case, a large amount of important, non-redundant information is omitted, resulting in significant visual artefacts, hence low BD rates. However, when using the MIV Main, all the non-redundant areas from all input views are included in atlases, making the synthesized final views presented to the user much more complete (see Figure 8).

Depending on the type of camera, the results for the MIV GA profile (right column of Table 1) can be divided into two parts. For perspective content, the Decoder Side Depth Estimation (DSDE) approach shows a significant reduction of coded bitrate for the same quality. In the MIV GA profile, geometry data is not coded, thus reducing the bitrate. Furthermore, by doing the depth estimation at the decoder, the MIV GA profile avoids destruction (e.g., blurring) of edges due to the low-quality of reconstructed depth maps at low bitrates. MIV GA can also efficiently encode multiview video even for CG content, even though the multiview simulcast approach has the advantage of having good quality input depth maps.

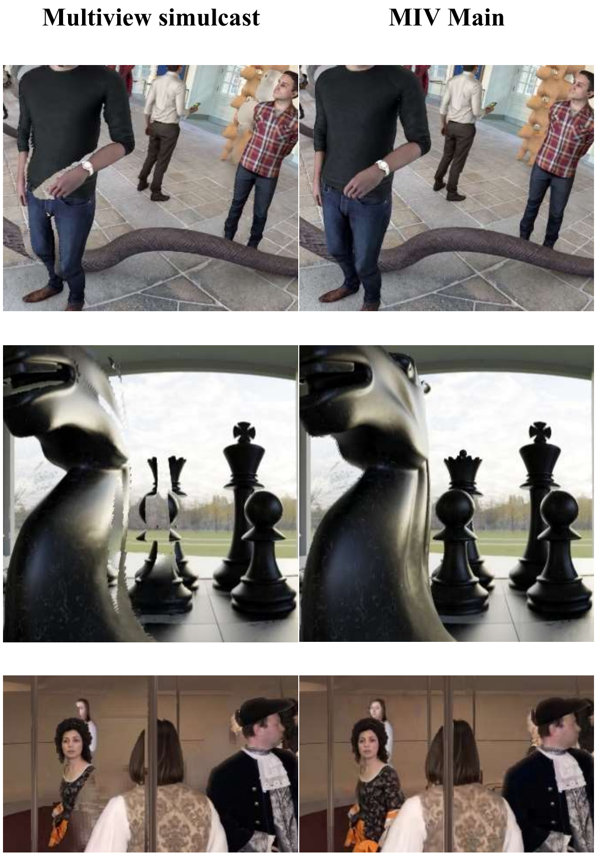


Figure 8: Subjective evaluation of the virtual view quality for two tested approaches (for each sequence, the total bitrate for MIV Main was not higher than bitrate for multiview simulcast); from top: Museum, Chess, Hijack.

For omnidirectional sequences, MIV Geometry Absent seems less effective than the multiview simulcast. However, such a result is not an effect of the MIV GA profile itself, but a weakness of the depth estimator used in the MIV-CTC, i.e., IVDE [24]. The current implementation of IVDE cannot generate high-quality depth maps for omnidirectional video.

Figure 9 contains results from Table 1, averaged over all test sequences of each content type. The orange RD-curves represent results for MIV Main, while grey curves correspond to the DSDE approach using the MIV Geometry Absent profile. The results of the multiview simulcast are shown as blue RD-curves.

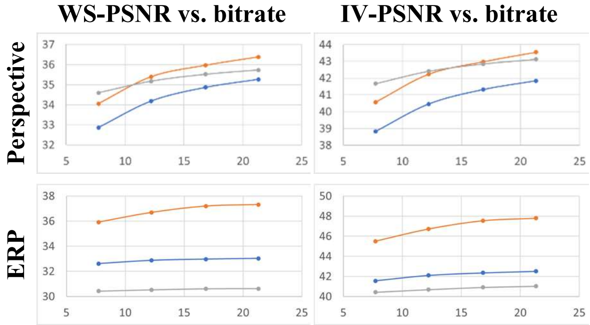
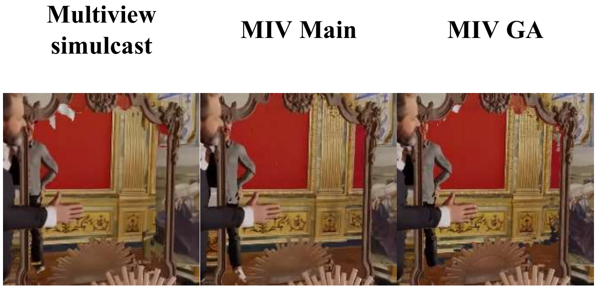


Figure 9: RD-curves for two classes of content, WS-PSNR and IV-PSNR in [dB], bitrate in [Mbps]; orange: MIV Main, grey: MIV GA, blue: multiview simulcast; WS-PSNR and IV-PSNR were averaged over all sequences within each class.

For perspective content, both profiles of MIV perform similarly and outperform the multiview simulcast approach. The MIV GA results are better than MIV Main at low bitrates because depth transmitted as geometry atlases in MIV Main suffers from high compression. For omnidirectional (ERP) sequences, the difference between MIV Main and the multiview simulcast is the highest, with 4 to 6 dB gain, proving its superiority in coding such content. High-quality results (more than 35 dB) are obtained with bitrates starting at 10 or 20 Mbps, depending on the sequence. The results of the MIV-GA profile are worse because the current implementation of IVDE cannot estimate good-quality depth maps for omnidirectional video.

Figure 10 shows a visual comparison for the three test approaches. For each test sequence, one frame of videos encoded at approximately the same bitrate are chosen for illustration. In most cases, MIV Main works the best, though in some cases MIV GA preserve edges better.



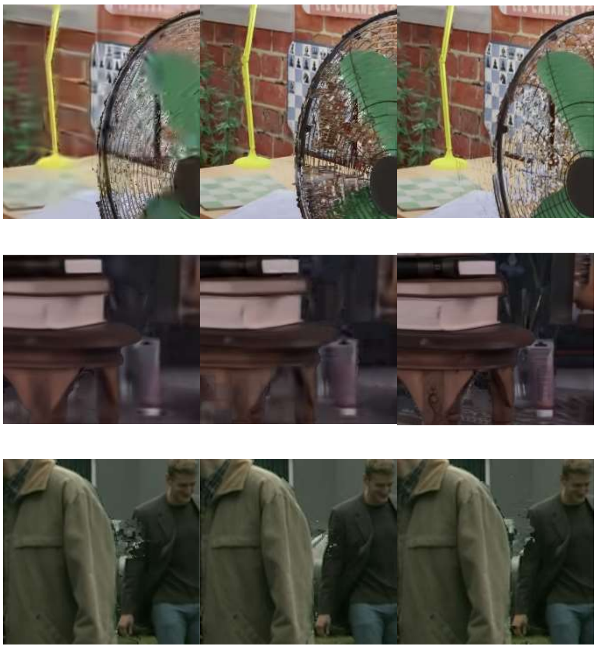


Figure 10: Subjective evaluation of the virtual view quality for three tested approaches (at approximately matched bitrate for each sequence – bitrate for each MIV approach did not exceed 107% of bitrate for multiview simulcast); from top: Mirror, Fan, Painter, Carpark.

The evaluation results provided in this section show a 4 to 6 dB compression gain compared to a naïve multiview simulcast approach. The results evaluating two of the three profiles of MIV edition-1 also demonstrates MIV's applicability in different use cases. The following section elaborates on aspects to be handled by the next edition of MIV.

BEYOND MIV EDITION-1

While MIV edition-1 focused on efficiently compressing immersive, dynamic volumetric video, there are opportunities to improve its flexibility to support new use cases. Particularly, additional facilities are required to support: (a) advanced camera settings, (b) handling surfaces that exhibit non-Lambertian characteristics, and (c) combining heterogenous input sources into a single bitstream. The document [25] provides a comprehensive list of new requirements and use cases that MIV edition-2 aims to address. The following subsections highlight some of the main ones.

Advanced camera settings

MIV Edition-1 supports camera arrays with intrinsics (e.g., the focal length) and extrinsics (the relative camera positions) that do not change in time. Furthermore, the colour and depth components are assumed to be captured from the same viewpoint; depth estimators that use computer vision techniques to estimate depth always comply with this constraint.

It is also possible to capture a 3D scene with multiple RGB-D cameras, like Kinect, which use different sensors to capture texture and depth. Due to physical constraints, the two sensors will have different poses. The multiple RGB-D cameras that capture the scene can also have varying intrinsic and extrinsic camera parameters that may change over time. MIV edition-2 will also include support for such heterogeneous camera rigs.

Non-Lambertian scenes

In estimating depth and performing view synthesis of novel viewpoints, there is often an implicit assumption that the colour of points on surfaces in the scene does not change with the viewing orientation. Such a scene is said to exhibit Lambertian reflectance characteristics. In practice, however, objects in the volumetric scenes very often have non-Lambertian reflectance characteristics, e.g., glossy, transparent, or highly reflective surfaces. For a surface that exhibits non-Lambertian characteristics, the geometry remains the same, but the appearance (texture) changes based on the orientation of the viewpoint. MPIs can approximate non-Lambertian surfaces for small viewing volumes well, but support for larger viewing volumes would need new extensions in MIV edition-2.

Furthermore, coding of scenes with non-Lambertian surfaces will also require algorithms to accurately identify and extract regions of the scene that exhibit such view-dependent light transport characteristics. The corresponding metadata should be efficiently compressed and added to the MIV bitstream to assist the renderer in reconstructing a novel viewport of the scene in a photorealistic manner. Some work [26] based on extending depth-image-based rendering has already been started as an exploration experiment.

Heterogeneous 3D scene representations

MIV edition-1 not only compresses and transmits multiview sources efficiently, but the standard also supports rendering the scene from any novel viewpoint within a pre-determined viewing space. However, in edition-1, the volumetric scene is rendered as is, without the ability to embed new volumetric objects or manipulate the pose of such embedded objects. These facilities are needed to support a metaverse use case. In this case, for example, the volumetric objects could be coded using a V-PCC bitstream. Embedding new volumetric objects into a MIV scene will then require additional signalling to map these objects from their local coordinate space to the space represented by the MIV scene.

Since MIV and V-PCC are extensions of the V3C data format, it is possible to code both multiview plus depth input sources and point cloud input sources into a single bitstream. Preliminary experiments [27] suggest this assumption is correct. Support for combining other input sources, including dynamic meshes, will also be studied as a part of MIV edition-2 activities.

CONCLUSION

This article introduced the MIV edition-1 standard, its intended use cases, and its place in the MPEG-I standard suite targeting immersive VR/XR applications. It provided an overview of the MIV encoding and decoding technologies that used intelligent data pruning and packing strategies. BD-PSNR coding gains of up to 6 dB in the 10 to 20 Mbps range are obtained, compared to a naive simulcast multiview plus depth video coding approach. MIV edition-1 is hence an evolution towards efficient immersive video coding technologies of the future.

Improving on MIV edition-1, MIV edition-2 will provide extensions supporting more flexibility to capture, code, and render immersive volumetric content. It will address the coding and rendering of non-Lambertian surfaces, often found in natural scenery. Heterogeneous data sources, like point clouds and meshes, coded and multiplexed into a single bitstream will also be supported.

REFERENCES

1. ISO/IEC 13818-2, Information technology — Generic coding of moving pictures and associated audio information — Part 2: Video.
2. ISO/IEC 14496-10, Information technology — Coding of audio-visual objects — Part 10: Advanced video coding.
3. ISO/IEC 23008-2, Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding.
4. ISO/IEC 23090-3, Information technology — Coded Representation of Immersive Media — Part 3: Versatile Video Coding.
5. B. Bross et al. “Overview of the Versatile Video Coding (VVC) standard and its applications,” IEEE Tr. on Circ. and Syst. for Vid. Tech., 2021.
6. ISO/IEC DIS 23090-12, Information technology — Coded Representation of Immersive Media — Part 12: MPEG immersive video.
7. D. Bonatto et al., “Real-Time Depth Video-Based Rendering for 6-DoF HMD Navigation and Light Field Displays,” IEEE Access, vol. 9, pp. 146868 – 146887, Oct. 2021.
8. J. Fleureau et al., "An Immersive Video Experience with Real-Time View Synthesis Leveraging the Upcoming MIV Distribution Standard," IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1-2, 2020.
9. S. Kwak et al., “View synthesis with sparse light field for 6DoF immersive video,” ETRI Journal Wiley, vol. 44(1), pp. 24–37, 2022.
10. “Reference view synthesizer (RVS) manual,” ISO/IEC JTC 1/SC29/WG11 N18068, 2018.
11. M. Wien et al., “Standardization status of immersive video coding,” IEEE J. Emerg. and Sel. Top. in Circ. and Syst., vol. 9, no. 9, Mar. 2019.
12. ISO/IEC DIS 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).
13. L. Kaufman and P.J. Rousseeuw, “Partitioning Around Medoids (Program PAM),” Wiley Series in Probability and Statistics, Hoboken, NJ, USA, pp. 68–125.
14. J. Jylänki, “A thousand ways to pack the bin—A practical approach to two-dimensional rectangle bin packing,” Tech. Rep., 2010, Online.
15. “Test Model 11 for MPEG Immersive video,” Doc. ISO/IEC JTC 1/SC 29/WG 04 N 0142, October 2021, Online
16. J. Boyce et al., “MPEG Immersive Video Coding Standard,” Proceedings of the IEEE, vol. 109 (9), 2021.
17. A. Wieckowski et al., “VVenC: an open optimized VVC encoder in versatile application scenarios,” Proc. SPIE 11842, App. of Digital Image Proc. XLIV, Aug. 2021.
18. D. Mieloch et al., “Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video,” IEEE Tr. on Circ. and Syst. for Vid. Techn., 2022
19. “Versatile view synthesizer (VVS) 2.0 manual,” ISO/IEC JTC 1/SC29/WG11 N18172, 2019.
20. “Common Test Conditions for MPEG Immersive Video,” ISO/IEC JTC 1/SC 29/WG04 N0169, Jan. 2022
21. Y. Sun et al, “Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video,” IEEE Signal Processing Letters 24.9(2017):1408-1412.
22. A. Dziembowski et al., “IV-PSNR – the objective quality metric for immersive video applications,” IEEE Tr. on Circ. and Syst. for Vid. Techn., 2022.
23. G. Bjoentegaard, “Calculation of average PSNR differences between RD-Curves,” ITU-T VCEG Meeting, Austin, USA, 2001.
24. “Manual of IVDE 3.0,” ISO/IEC JTC1/SC29/ WG4 MPEG VC/N0058, Jan. 2021.
25. “Use cases and requirements for MIV – edition-2 (final).”, ISO/IEC JTC 1/SC 29/WG02 N0157, Jan. 2022.
26. S. Fachada et al., “Depth Image-Based Rendering of Non-Lambertian Content in MPEG Immersive Video,” 2021 Int. Conf. on 3D Immersion (IC3D), pp. 1-6, Dec. 2021.
27. Z. Zhu and L.Yu, Report on the progress of V3C Bitstream with Heterogeneous Sources, ISO/IEC JTC 1/SC 29/WG 4 m 59559, April 2022.

Vinod Kumar Malamal Vadakital, is a Principal Scientist at Ofinno LLC, Tampere, Finland. His research interests include video signal processing, computer vision, and XR technologies. Vinod Kumar Malamal

Vadakital received the Ph.D. degree in signal processing from Tampere University of Technology. Contact him at vinod.malamalvadakital@ofinno.com.

Adrian Dziembowski is an Assistant Professor with the Institute of Multimedia Telecommunications, Poznań University of Technology, Poznań. He received the Ph.D. degree from the Poznań University of Technology in 2018. He authored or co-authored over 40 papers on various aspects of immersive video, free navigation, and free-viewpoint television systems. He is also actively involved in ISO/IEC MPEG activities towards MPEG Immersive video coding standard. Contact him at adrian.dziembowski@put.poznan.pl.

Gauthier Lafruit is an Associate Professor of immersive light field technologies at Université Libre de Bruxelles (ULB), Brussels, Belgium. He received the M.Sc. and Ph.D. degrees from Vrije Universiteit Brussel (VUB), in 1989 and 1995, respectively. He works in visual data compression and rendering, participating in compression standardization committees like CCSDS (space applications), JPEG (still picture coding) and MPEG (moving picture coding). His research interests focus on depth image-based rendering, immersive video, and digital holography. Contact him at gauthier.lafruit@ulb.be.

Franck Thudor is a Researcher at InterDigital, Rennes, France. His current research interests include immersive experience, video compression, and he is particularly involved in ISO/IEC 23090-12 MPEG immersive video standardization. He received an engineering degree from Ecole Nationale Supérieure des Télécommunications de Bretagne, France, in 1999**.** Contact him at franck.thudor@interdigital.com.

Gwangsoon Lee, is a Principal Researcher in Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include immersive video coding, light field image processing and realistic video system. Gwangsoon Lee received his Ph.D. degree in Electronics Engineering from Kyungpook National University, Daegu, South Korea. Contact him at gslee@etri.re.kr.

Patrice Rondao Alface, is a Senior Research Engineer at Nokia Technologies, Antwerp, Belgium. His research interests include the Volumetric Video Coding of mesh, point cloud and immersive video data. Patrice Rondao Alface received the Ph.D. degree in Applied Sciences from Universite catholique de Louvain, Belgium. He is Senior Member of the IEEE Signal Processing Society. Contact him at patrice.rondao_alface@nokia.com.