INVITED PAPER

An Overview of Omnidirectional MediA Format (OMAF)

The OMAF standard is arguably the first virtual reality (VR) system standard that includes support for 360° video (among others). This article introduces the end-to-end OMAF architecture from content authoring to the player and highlights representation formats of omnidirectional video and images.

By Miska M. Hannuksela, *Member IEEE*, and Ye-Kui Wang

ABSTRACT | During recent years, there have been product launches and research for enabling immersive audio–visual media experiences. For example, a variety of head-mounted displays and 360° cameras are available in the market. To facilitate interoperability between devices and media system components by different vendors, the Moving Picture Experts Group (MPEG) developed the Omnidirectional MediA Format (OMAF), which is arguably the first virtual reality (VR) system standard. OMAF is a storage and streaming format for omnidirectional media, including 360° video and images, spatial audio, and associated timed text. This article provides a comprehensive overview of OMAF.

KEYWORDS | 360° video; Dynamic Adaptive Streaming over HTTP (DASH); file format; Omnidirectional MediA Format (OMAF); omnidirectional media; viewport; virtual reality (VR).

I. INTRODUCTION

Virtual reality (VR) has been researched and trialed for many years [1], [2]. Due to the growth of computing capability in devices and network bandwidth, as well as advances in the technology for head-mounted displays (HMDs), wide deployment of VR became possible only recently. Facebook's two-billion-dollar acquisition of Oculus in 2014 seemed to be a start and a catalyst to the fast proliferation of VR research and development, device production, and services throughout the globe. Almost suddenly, VR became a buzzword everywhere in the world, many companies in the information and communication technology field started to have VR as an important strategic direction, and all kinds of VR cameras and devices started to be available in the market.

Unavoidably, numerous, different, noninteroperable VR solutions have been designed and used. This called for standardization, for which the number one target is always to enable devices and services by different manufactures and providers to interoperate.

The Moving Picture Experts Group (MPEG) started to look at the development of a VR standard in October 2015. This effort led to the arguably first VR system standard, called Omnidirectional MediA Format (OMAF) [3]. OMAF defines a media format that enables omnidirectional media applications, focusing on 360° video, images, and audio, as well as the associated timed text. The first edition (also referred to as the first version or v1) of OMAF was finalized in October 2017. It provides basic support for 360° video, images, and audio with three degrees of freedom (3DOF), meaning that only rotations around any coordinate axes are supported. Since the finalization of the standard, source code packages of several implementations compatible with OMAF v1 have been made publicly available [4]–[6]. The development of the second edition of OMAF was completed in October 2020. OMAF v2 [7] includes all v1 features and also supports richer 360° presentations with overlays and multiple viewpoints and improves viewport-dependent delivery. OMAF v2 enables limited support for six degrees of freedom (6DOF), where the translational movement of the user impacts the rendering of overlays. Even though OMAF v2 was just recently finalized, there are already implementations supporting its new features [8], [9].

Manuscript received February 28, 2020; revised October 29, 2020; accepted February 19, 2021. Date of publication March 17, 2021; date of current version August 20, 2021. *(Corresponding author: Miska M. Hannuksela.)*
Miska M. Hannuksela is with Nokia Technologies, 33100 Tampere, Finland (e-mail: miska.hannuksela@nokia.com).
Ye-Kui Wang is with Bytedance Inc., San Diego, CA 92130 USA (e-mail: yekui.wang@bytedance.com).

Digital Object Identifier 10.1109/JPROC.2021.3063544

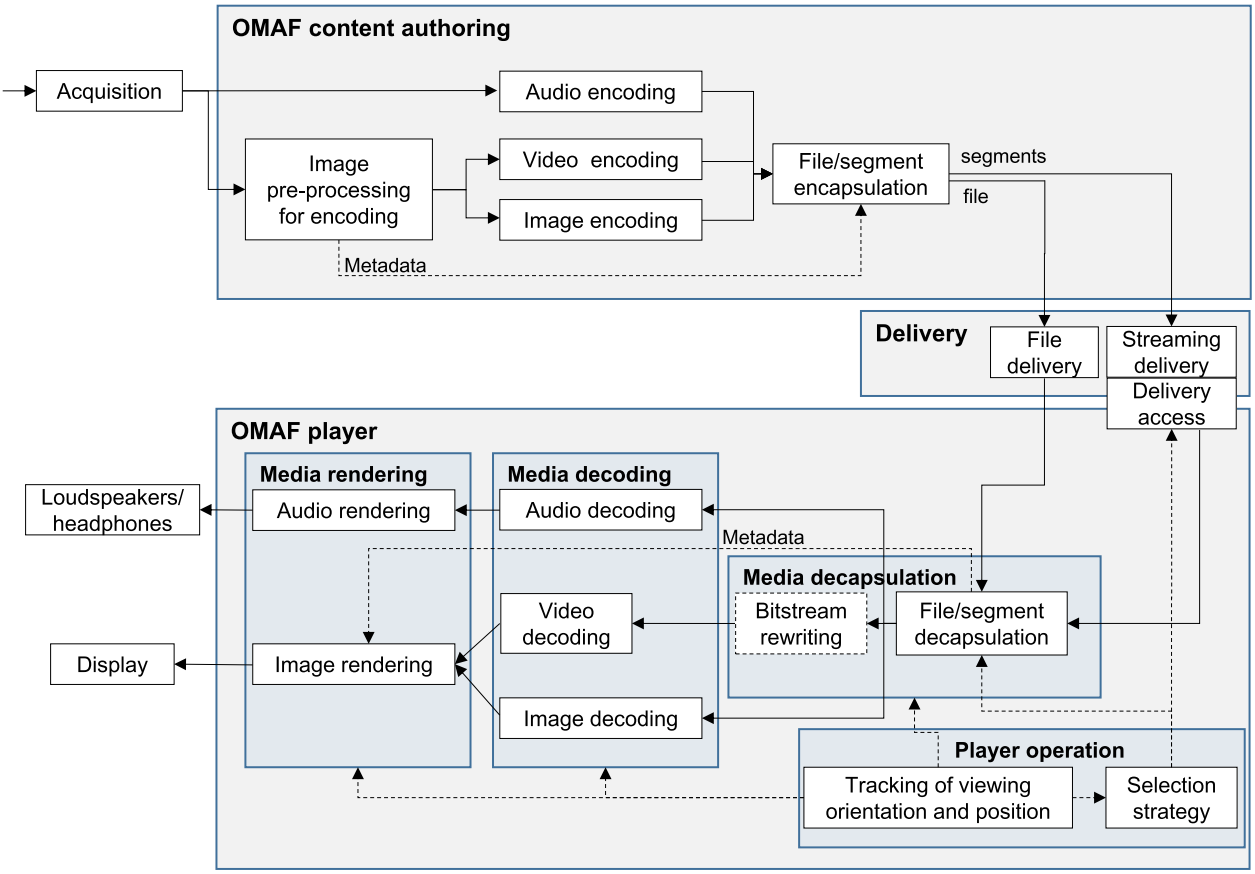

Fig. 1. *OMAF architecture.*

OMAF has been further profiled to suit specific industries and environments by the VR Industry Forum (VRIF) and the 3rd Generation Partnership Project (3GPP). VRIF has the mission to advocate industry consensus on standards for the end-to-end VR ecosystem and chose to reference some of the OMAF media profiles and specific viewport-dependent streaming scenarios in the VRIF Guidelines [10]. Just a few months after finalizing OMAF v2, the VRIF Guidelines were updated to version 2.3, which incorporates selected video profiles and toolset brands from OMAF v2. At the time of writing this article, the Streaming Video Alliance is carrying out a trial using selected OMAF video profiles as recommended in the VRIF Guidelines for streaming to various end-user devices [11]. 3GPP standardizes cellular telecommunications, including multimedia services. The 3GPP specification on VR profiles for streaming applications [12] is based on technical elements specified in OMAF v1.

Fig. 1 shows the OMAF architecture, which consists of three major modules: OMAF content authoring, delivery, and OMAF player. The OMAF content authoring module consists of media acquisition, omnidirectional video/image preprocessing, media encoding, and media file and segment encapsulation. OMAF may either use file delivery or streaming delivery for which the content is timewise partitioned into segments. The OMAF player module mainly consists of the media file and segment decapsulation, media decoding, and media rendering. In some operation modes, the media decapsulation block may contain a bitstream rewriting process that combines several delivered streams into one video bitstream for decoding and rendering. Note that the rendering process is not normatively specified in the OMAF standard. The OMAF player also contains essential processing blocks for the player operation, namely, the tracking and selection strategy modules. The tracking module controls the viewing orientation and, in some cases, also the viewing position according to which the content is rendered. For example, the tracking module may obtain the head orientation when an HMD is used for rendering. The selection strategy module makes the decisions that content pieces are streamed. The delivery access module acts as a bridge between the selection strategy and stream(s) delivery.

The media types supported in OMAF include video, audio, image, and timed text. However, in this article, we focus only on video and image, and therefore, we will not discuss audio and timed-text beyond this point.

The key underlying technologies for file/segment encapsulation and delivery of OMAF are ISO Base Media File Format (ISOBMFF) [13] and Dynamic Adaptive

Streaming over HTTP (DASH) [14]. OMAF specifies file format and DASH extensions in a backward-compatible manner, which enables reusing of existing ISOBMFF and DASH implementations for conventional 2-D media formats with only moderate changes. Note that, while OMAF also specifies signaling and delivery of omnidirectional media over MPEG Media Transport (MMT, ISO/IEC 23008-1), it is not discussed in this article.

This article is organized as follows. ISOBMFF and DASH basics are reviewed in Section II. Representation formats of omnidirectional video/image are discussed in Section III. Section IV provides an introduction to 360° video streaming with an emphasis on viewport-dependent streaming, which mitigates the large resolution and high bitrate required for 360° video by prioritizing the displayed area, i.e., the viewport. Sections V and VI present the OMAF video and image profiles, which specify how a media codec is adapted for omnidirectional application usage. OMAF v2 defines the concept of toolset brands for functionalities beyond basic playback of omnidirectional audio–visual content. Toolset brands are elaborated in Section VII. In Section VIII, we draw a conclusion and take a look at future VR standardization work in MPEG.

This article contains a significant amount of additional details compared to our earlier paper that provides a simpler overview of OMAF v1 [15]. Furthermore, we have added the descriptions for omnidirectional images and OMAF image profiles. Moreover, this article is arguably the first publication that provides a comprehensive review of OMAF v2.

II. BACKGROUND

A. ISOBMFF and HEIF

The ISOBMFF is a popular media container format for audio, video, and timed text. ISOBMFF compliant files are often casually referred to as MP4 files. The High Efficiency Image File Format (HEIF) [16] derives from the ISOBMFF and is gaining popularity as a storage format for still images and image sequences, such as exposure stacks. It is natively supported by major operating systems for smartphones and personal computers, i.e., iOS and Android, as well as Windows 10 and MacOS. OMAF file format features for omnidirectional video and still images are built on top of ISOBMFF and HEIF, respectively.

A basic building block in ISOBMFF is called a box, which is a data structure consisting of a four-character-code (4CC) box type, the byte count of the box, and a payload, whose format is determined by the box type and which may contain other boxes. An ISOBMFF file consists of a sequence of boxes.

Each stream of timed media or metadata is logically stored in a track, for which timestamps, random access positions, and other information are provided in respective boxes. The media data for tracks are composed of samples carried in `MediaDataBox`(es), where each sample corresponds to the coded media data of a single time instance. It is possible to store the track metadata for its entire duration in a `MovieDataBox` or split the metadata in time ranges using `MovieFragmentBox`es. In a self-contained movie fragment, the `MediaDataBox` containing the samples of a movie fragment is next to the respective `MovieFragmentBox`.

A sample entry of a track describes the coding and encapsulation format used in the samples and includes a 4CC sample entry type and contained boxes that provide further information of the format or content of the track. A restricted video sample entry type ("`resv`") is used for video tracks that require postprocessing operations after decoding to be displayed properly. The type of postprocessing is specified by one or more scheme types associated with the restricted video track.

ISOBMFF defines items for storing untimed media or metadata, and HEIF uses items for storing still images. In addition to coded image items, HEIF supports derived image items, where an operation corresponding to the type of the derived image item is performed to one or more indicated input images to produce an output image to be displayed. The "`grid`" derived image item arranges input images onto a grid to create a large output image. Metadata that are specific to an item are typically stored as an item property. A comprehensive technical summary on HEIF is available in [17].

B. DASH

DASH specifies a Media Presentation Description (MPD) format for describing the content available for streaming and segment formats for the streamed content. There are three basic types of segments in DASH: initialization segment, media segment, and index segment. Initialization segments are meant for bootstrapping the media decoding and playback. Media segments contain the coded media data. Index segments provide a directory to the media segments for accessing them in a more fine-grained manner than on a segment basis. In the segment format for ISOBMFF, each media segment consists of one or more self-contained movie fragments, whereas the movie header containing the track header is delivered as an initialization segment. It is possible to omit separate initialization segments by creating self-initializing media segments that contain the necessary movie and track headers. Conventionally, index segments have not been used with ISOBMFF, but rather each media segment can be split into subsegments that are indexed within the media segment itself. DASH does not specify carriage of image items, but, since an image item can be used as a viewpoint, an overlay, or background for overlays, OMAF v2 specifies carriage of image items as self-initializing media segments. Fig. 2 summarizes how timed and static media are encapsulated into ISOBMFF files and further into segments for DASH delivery.

Conventionally, DASH can be used in two operation modes, namely, live and on-demand. For both operation modes, the DASH standard provides profiles that specify

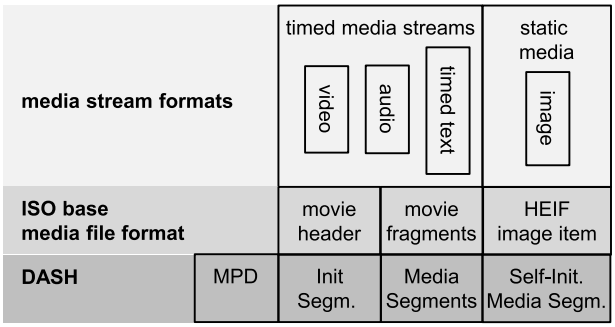

Fig. 2. *Relation of media stream formats, ISOBMFF, and DASH units.*

constraints on the MPD and segment formats. In the live profiles, the MPD contains sufficient information for requesting media segments, and the client can adapt the streaming bitrate by selecting the representations from which the media segments are received. In the on-demand profiles, in addition to the information in the MPD, the client typically obtains an index of subsegments of the media segments of each representation. The client selects the representation(s) from which subsegments are fetched and requests them using byte-range requests.

The MPD syntax is specified as an Extensible Markup Language (XML) schema and contains one or more adaptation sets, each containing one or more representations. A representation corresponds to an ISOBMFF track, and an adaptation set contains representations of the same content between which the player can select, e.g., based on the available bitrate.

The MPD format includes bitrates and other characteristics for representations and adaptation sets for player-driven content selection. DASH specifies essential and supplemental property descriptor elements for describing additional characteristics of representations or adaptation sets. When a player does not recognize an essential property descriptor, it is required to omit the representation or adaptation set that contains the descriptor. In contrast, a player is allowed to ignore an unknown supplemental property descriptor and continue the processing of the respective representation or adaptation set.

An MPD contains either a template for deriving a uniform resource locator (URL) for each segment or a list of segment URLs. Players use the URLs (or byte ranges of them) of the selected segments when requesting the content over the Hypertext Transfer Protocol (HTTP). A conventional web server can be used for responding to HTTP requests.

III. REPRESENTATION FORMATS OF OMNIDIRECTIONAL VIDEO AND IMAGES

A. Introduction

OMAF specifies three types of representation formats, namely, projected, mesh, and fisheye omnidirectional video and images. These formats differ in image prepreprocessing for encoding, in the arrangement of visual content in both the input pictures (for encoding) and the decoded pictures, and in the image rendering processing blocks. A summary of the omnidirectional video and image representation formats is provided in Table 1. In all representation formats, both monoscopic and stereoscopic contents are allowed, and the content coverage can be less than 360°.

This section is organized as follows. Section III-B discusses the coordinate systems used in OMAF. Sections III-C and III-D describe the projection formats and regionwise packing (RWP), respectively. Sections III-E and III-F present the mesh and fisheye omnidirectional formats, respectively. Finally, Section III-G provides a brief review of supplemental metadata for omnidirectional video and images.

B. Coordinate Systems

As illustrated in Fig. 3, the OMAF coordinate system consists of a unit sphere and three coordinate axes. The location of a point on the sphere is identified by a pair of sphere coordinates azimuth (ϕ) and elevation (θ). The user looks from the center of the sphere outward toward the inside surface of the sphere.

OMAF specifies global coordinate axes that are shared for all media types intended to be rendered together and used for determining the initial viewing orientation. Each video or image may use its own local coordinate axes specified by the *X*-, *Y*-, and *Z*-axes of the coordinate system after the application of a rotation to the global coordinate axes, where the rotation consists of yaw, pitch, and roll rotation angles, around the *Z*-, *Y*-, and *X*-axes, respectively. The use of unaligned global and local coordinate axes can be advantageous, e.g., for correcting the horizon to be exactly horizontal in the projected omnidirectional video or image or for improving perceived picture quality by avoiding seams between projection surfaces to cross objects of interest. OMAF specifies the signaling and the

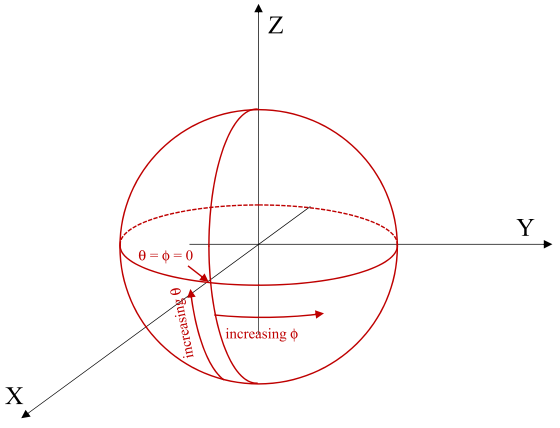

Fig. 3. *OMAF coordinate system [3].*

Table 1 Summary of Omnidirectional Video and Image Representation Formats in OMAF

	Pre-processing for encoding				Rendering
Projected omnidirectional video/images	Stitching captured images onto a spherical image	Rotation from global to local coordinate axes (optional)	Projection using a mathematically specified projection format	Rectangular region-wise packing (optional)	Mapping regions of the decoded pictures (if region-wise packing was applied) or the entire projected picture (otherwise) onto a rendering mesh suitable for the projection format in use
Mesh omnidirectional video			Projection onto a 3-D mesh of mesh elements	Arranging mesh elements onto 2-D picture(s)	Mapping mesh elements from the decoded 2-D picture(s) to the 3-D mesh, which is used directly as the rendering mesh.
Fisheye omnidirectional video/images				Arranging captured circular images onto a 2-D picture	Stitching the decoded circular images onto a spherical image and further onto a suitable rendering mesh.

rotation equations for the conversion between the global coordinate system and a local coordinate system.

C. Omnidirectional Projection Formats

Omnidirectional projection is a necessary geometric operation applied at the content production side to generate 2-D pictures from the stitched sphere signal, and an inverse projection operation needs to be used in the rendering process by the OMAF player.

OMAF specifies the support of two types of projection: equirectangular projection (ERP) and cubemap projection (CMP). In addition to ERP and CMP, a number of other projection methods were studied during the OMAF v1 standardization process, but none of them were found to provide sufficient technical benefits over the widely used ERP and CMP formats.

As illustrated in Fig. 4, the ERP process is close to how a 2-D world map is typically generated, but with the left-hand side being the east instead of the west, as the viewing perspective is opposite. In ERP, the user looks from the center of the sphere outward toward the inside surface of the sphere, while, for a world map, the user looks from outside the sphere toward the outside surface of the sphere.

As illustrated in Fig. 5, in the CMP specified in OMAF, the sphere signal is rectilinearly projected into six square faces that are laid out to form a rectangle with a 3:2 ratio of width versus height, with some of the faces rotated to maximize continuity across face edges.

D. Regionwise Packing

RWP is an optional step after projection on the content production side. It enables resizing, repositioning, rotation by 90°, 180°, or 270°, and vertical/horizontal mirroring of any rectangular region before encoding.

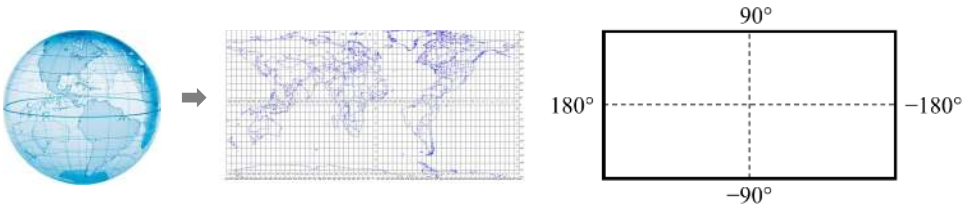

Fig. 4. *Illustration of the ERP.*

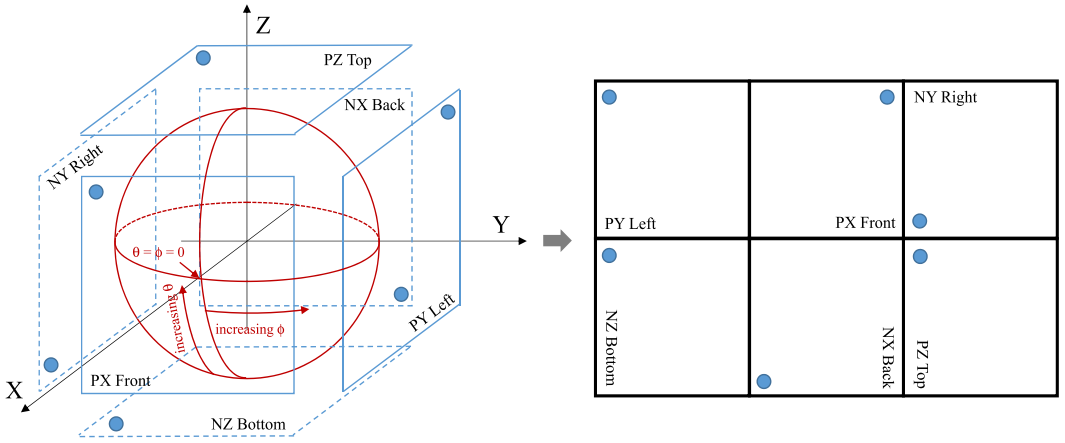

Fig. 5. *Illustration of the CMP in OMAF [3].*

RWP can be used, e.g., for the following purposes: 1) indicating the exact coverage of content that does not cover the entire sphere; 2) generating viewport-specific (VS) video or extractor tracks with region-wise mixed-resolution packing or overlapping regions; 3) arranging the cube faces of CMP in an adaptive manner; 4) providing guard bands by adding some additional pixels at geometric boundaries when generating the 2-D pictures for encoding, which can be used to avoid or reduce seam artifacts in rendered 360° video due to projection, and 5) compensating the oversampling of pole areas in ERP.

An example of using RWP for compensating the oversampling of pole areas in ERP is presented in Fig. 6. First, an ERP picture is split into three regions: top, middle, and bottom, where the top and bottom regions cover the two poles and have the same height, while the middle region covers the equator. Second, the top and bottom regions are subsampled to keep the same height but half of the width, and then, the subsampled top and bottom regions are placed next to each other on top of the middle region. This way, the equator area remains the same resolution, while the top and bottom regions got subsampled to half of the width, which compensates for the oversampling of the pole areas in ERP.

The RWP metadata indicate the interrelations between regions in the projected picture (e.g., an ERP picture) and the respective regions in the packed picture (i.e., the picture in the coded video bitstream) through the position and size of the regions in both projected and packed pictures, as well as indications of the applied rotation and mirroring, if any. When RWP has been applied, the decoded pictures are packed pictures characterized by RWP metadata. Players can map the regions of decoded pictures onto projected pictures and, consequently, onto the sphere by processing the RWP metadata.

E. Mesh Omnidirectional Video

OMAF v2 adds the 3-D mesh format as a new omnidirectional content format type. A 3-D mesh is specified as a set of mesh elements, all of which are either parallelograms or regions on a sphere surface. The parallelograms can appear at any location and orientation within the unit sphere and need not be connected. A sphere-surface mesh element is specified through an azimuth range and an elevation range, as illustrated in Fig. 7. Thus, it is possible to specify a 3-D mesh to represent both ERP and CMP as special cases. However, the mesh omnidirectional video provides flexibility for optimizing the projection beyond ERP and CMP.

The given 3-D mesh can be used directly for rendering. In other words, the 3-D mesh format enables direct one-to-one mapping of regions of a 2-D image to elements of a 3-D mesh, which is often referred to as UV mapping in computer graphics terminology. The 3-D mesh format avoids the need for deriving the UV map according to the projection format and the RWP metadata.

F. Fisheye Omnidirectional Video and Images

Fisheye video/images do not use projection or RWP. Rather, for each picture, the circular images captured by fisheye cameras are directly placed onto a 2-D picture, e.g., as shown in Fig. 8.

Parameters indicating the placement of the circular images on the 2-D picture and the characteristics of the fisheye video/images are specified in OMAF and can be used for correct rendering. The fisheye format avoids the need for real-time stitching in video recording. OMAF files with fisheye video/images could be suitable for low-cost consumer 360° cameras and smartphones, for example.

G. Supplemental Metadata for Omnidirectional Video and Images

This section provides a summary of supplemental metadata for omnidirectional video or images that may optionally be present in OMAF files or MPDs.

Regionwise Quality Ranking (RWQR): OMAF specifies RWQR metadata as a basic mechanism to enable viewport-dependent content selection. Quality ranking metadata can be provided for sphere regions and for rectangular regions on decoded 2-D pictures. Quality ranking values are given for indicated regions and describe the relative

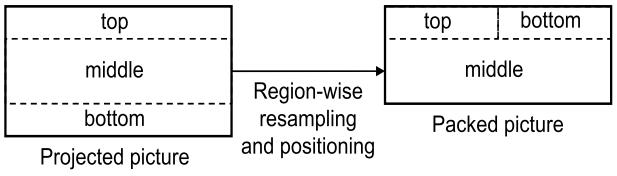

Fig. 6. *Example of using RWP for compensating pole area oversampling of ERP.*

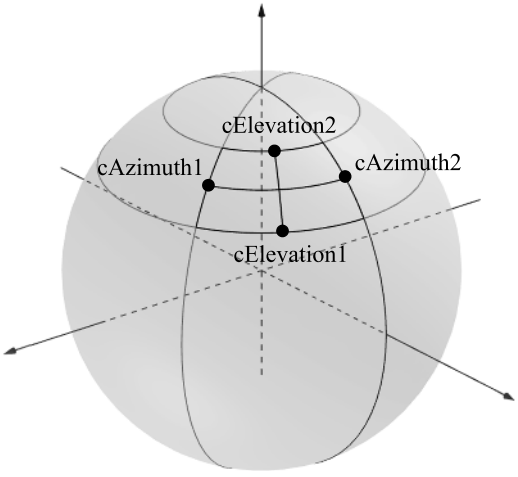

Fig. 7. *Mesh element specified as a region on the sphere surface through an azimuth range and an elevation range.*

Fig. 8. *Example fisheye omnidirectional video captured by two lenses.*

quality order of the regions: when region A has a nonzero quality ranking value less than that of region B, region A has a higher quality than region B. RWQR metadata remain static for the entire duration of the track. OMAF players can use RWQR metadata for a viewport-dependent selection of tracks for streaming and/or playback.

ERP region timed metadata provides a time-varying relative quality rank recommendation, relative priority information, or heatmap signaling for a rectangular grid relative to ERP. OMAF players may use the information for spatially fine-grained streaming rate adaptation choices so that picture quality is first reduced in regions that are subjectively the least important.

Initial Viewing Orientation: The default viewing orientation to start displaying the omnidirectional video or image is along the *X*-axis of the global coordinate axes. Content authors can override the default behavior by using an initial viewing orientation timed metadata track and item property for video and images, respectively. If an HMD is used for viewing, players are expected to obey only the indicated initial azimuth. Otherwise (i.e., when a conventional 2-D display is used for viewing), players should use the initial azimuth, elevation, and tilt for rendering. Initial viewing orientation can be indicated to apply also during normal playback. This is helpful to reset the viewing orientation toward to content author's choice after a scene cut.

Recommended Viewport Timed Metadata: OMAF supports a playback mode where a user does not have or has given up control of the viewing orientation. Such usage may suit for example displaying omnidirectional video on a conventional flat-panel display. Rather than the user controlling the viewing orientation, the displayed viewport is indicated in a recommended viewport timed metadata track. Several recommended viewport tracks can be made available, may be indicated to be based on viewing statistics or manual selections, and may be labeled with a description.

The 2-D spatial relationship track grouping provides another option for viewport-dependent omnidirectional video streaming, in addition to the viewport-dependent video profiles. Each track in an indicated 2-D spatial relationship group corresponds to a planar spatial part of a video source. The signaling indicates the size (width and height) of the original video content and the position and size of each of the split sub-pictures. In addition, the signaling also indicates whether a sub-picture track is intended to be presented alone without any other sub-picture tracks from the same original video content and whether video bitstream carried in the sub-picture track can be merged with the video bitstream carried in any other sub-picture tracks split from the same original video content to generate a single video bitstream without decoding mismatch by rewriting only the header data of the bitstreams, where a decoding mismatch refers to the value of any pixel when decoding the video bitstream in the current track is not identical to the value of the same pixel when decoding the merged video bitstream. Besides file format signaling for sub-pictures, OMAF v2 also specifies DASH signaling for sub-pictures through the sub-picture composition identifier element, which indicates the DASH adaptation sets that contain sub-picture representations carrying sub-picture tracks belonging to the same 2-D spatial relationship track group.

IV. 360° VIDEO STREAMING

A. Introduction

This section reviews approaches for omnidirectional video streaming and describes which building blocks OMAF provides for them. Section V describes further details on the types and features of 360° video streaming that are supported in OMAF video profiles.

360° video streaming can either be carried out in a viewport-independent or viewport-dependent manner. In viewport-independent 360° video streaming, no picture quality emphasis is given to any spatial part of the video, and the prevailing viewing orientation has no impact on which version of the video content is streamed. However, since the spherical sampling density depends on the elevation angle in the ERP format, content authoring for ERP may be adapted to provide a more consistent picture quality in the spherical domain with any approach described in Table 2. Typically, a sequence of projected omnidirectional pictures is encoded in one or more bitrate or resolution versions, each of which is made available for streaming as a single DASH representation. A client selects the version that best suits its display resolution and the prevailing throughput.

Since the viewport covers only a fraction of the omnidirectional video at any time instance, a large portion of the omnidirectional video is not displayed. Thus, network bandwidth is inefficiently utilized in viewport-independent 360° video streaming. A key idea of viewport-dependent 360° video streaming is to dedicate a large share of the available bandwidth for the video covering the viewport. Studies presented in [24]–[26] have shown that viewport-dependent streaming is able to reach a bit rate reduction of several tens of percents compared to viewport-independent streaming. Since there is an

Table 2 Approaches for Improving the Compression of ERP Video for Viewport-Independent Streaming

Approach	Description	References
Latitude-dependent low-pass filtering	ERP images are low-pass filtered prior to encoding. The strength of the filter is a function of the latitude, providing the strongest smoothing on the poles.	[20]
Regional down-sampling	ERP images are split into stripes, which are horizontally downsampled except the stripe covering the equator. The stripes are then packed into a rectangular image for encoding.	[21]
Latitude-dependent quantization	The quantization parameter used in encoding is adapted according to the latitude. Thus, the transform coefficients are quantized more coarsely on the poles than in the equator.	[22]
Spherical-domain rate-distortion optimization	The distortion metric used in rate-distortion optimized encoding is adapted so that its impact in spherical domain remains approximately constant. The lambda for rate-distortion optimization and quantization parameter are also adapted accordingly.	[23]

inherent delay in the streaming system to react to viewport changes, the spherical video not contained within the viewport is typically streamed too albeit at a lower bitrate and thus also at lower picture quality. Another benefit provided by some viewport-dependent streaming approaches over viewport-independent streaming is that the sample count can be nonuniformly allocated, with a higher sampling density covering the viewport. Thus, the effective resolution on the viewport is greater than what the decoding capacity would otherwise support. An example scheme where the content of the viewport originates from a 6K (6144 $\times$ 3072) ERP was presented in [27].

One approach for viewport-dependent streaming is to create multiple VS 360° streams by encoding the same input video content for a predefined set of viewport orientations. Each stream also covers areas other than the targeted viewport, though at lower quality. Moreover, the content may be encoded for several bitrates and/or picture resolutions. The streams are made available for streaming, and metadata describing the viewports that the streams are aimed for are provided. Clients select the 360° stream that is targeted for their current viewport and suits the network throughput. Approaches to achieve VS 360° streams are summarized in Table 3.

In tile-based viewport-dependent 360° streaming, projected pictures are encoded as several tiles. Early approaches, such as [29] and [30], split the video prior to encoding into regions that were encoded independently of each other and decoded with separate decoding instances. However, managing and synchronizing many video decoder instances pose practical problems. Thus, a more practical approach is to encode tiles in a manner that they can be merged to a bitstream that can be decoded with a single decoder instance. Thus, in the context of viewport-dependent 360° streaming, the term tile commonly refers to an isolated region [31], which depends only on the collocated isolated region in reference pictures and does not depend on any other picture regions. Several versions of the tiles are encoded at different bitrates and/or resolutions. Coded tile sequences are made available for streaming together with metadata describing the location of the tile on the omnidirectional video. Clients select which tiles are received so that the viewport has higher quality and/or resolution than the tiles outside the viewport. A categorization of tile-based viewport-dependent 360° streaming is presented in Table 4.

The remaining part of this section discusses tile-based viewport-dependent streaming and is organized as follows. The present OMAF video profiles use either the Advanced Video Coding (AVC) [18] or the High Efficiency Video Coding (HEVC) [19] standard as the basis. Section IV-B describes the use of AVC and HEVC for tile-based viewport-dependent streaming. In a typical arrangement for tile-based viewport-dependent 360°, a player binds received tiles into a single video bitstream for decoding. Section IV-C presents tile binding approaches applicable to OMAF video profiles. Section IV-D introduces tile index and tile data segment formats that are specified in OMAF v2 for improving viewport-dependent streaming. Section IV-E discusses a content authoring pipeline for tile-based viewport-dependent streaming.

B. Isolated Regions in AVC and HEVC

Video coding formats provide different high-level structures for realizing isolated regions, which are used as elementary units in tile-based viewport-dependent 360° streaming. This section provides more details on how isolated regions can be realized in AVC and HEVC.

Table 3 Approaches for Achieving VS 360° Streams

Approach	Description	References
VS low-pass filtering	Areas that are outside the viewport are low-pass filtered prior to encoding	[28]
Asymmetric projection	An asymmetric projection format is used. For example, a spherical picture can be projected onto a regular square pyramid where the viewport is covered by the base of the pyramid.	[25]
VS region-wise mixed-resolution packing	A symmetric projected picture is region-wise resampled and packed	[25]
VS variable-quality encoding	The area covered by viewport is coded with higher quality/bitrate by adjusting the quantization parameter spatially	OMAF Annex D

Table 4 Tile-Based Viewport-Dependent 360° Streaming Approaches [15]

Approach	Description	References
Region-wise mixed quality (RWMQ)	Several versions of the content are coded with the same tiling, each with different bitrate and/or picture quality. The resolution of all the versions is the same, and typically the tile sizes are identical regardless of their location within the picture. Players choose high-quality tiles to cover the viewport and low-quality tiles covering the remaining parts of the sphere.	[24], OMAF Annex D
Viewport + 360° video	A complete low-resolution and/or low-quality omnidirectional picture is encoded. Tiling may but need not be used in this low-resolution/low-quality version. Additionally, tiled content is coded at high resolution and/or picture quality. Players receive the low-resolution/low-quality version and the high-resolution/high-quality tiles covering the viewport.	[21], OMAF Annex D
Region-wise mixed resolution (RWMR)	Tiles are encoded at multiple resolutions. Tile partitioning is typically such that tiles are not overlapping on the sphere. Players select a combination of high-resolution tiles covering the viewport and low-resolution tiles for the remaining areas.	[27], [32], OMAF Annex D

In HEVC, a picture is split into tiles along a grid of tile columns and rows. A slice can be either an integer number of complete tiles or a subset of a single tile. Coded slices consist of a slice header and slice data. Among other things, the slice header indicates the position of the slice within the picture. Encoders can choose to use only rectangular slices, keep the tile and slice boundaries unchanged throughout a coded video sequence, and constrain the coding mode and motion vector selection so that a slice references only the collocated slices in the reference picture(s). In a common operation mode, a slice encloses a set of one or more complete tiles, which can be referred to as a motion-constrained tile set (MCTS).

AVC does not enable picture partitioning into tiles. However, slices can be arranged vertically into a single column, and their encoding can be constrained as described above for HEVC.

A sub-picture is a picture that represents a spatial subset of the original video content. Consequently, a sub-picture bitstream represents a sub-picture sequence. As an alternative to partitioning pictures into tiles and/or slices, pictures can be split prior to encoding into sub-picture sequences. Each sub-picture sequence is encoded with constraints in the coding modes and motion vectors so that the encoded sub-picture bitstreams can be merged into a single bitstream with multiple tiles.

Each coded tile or sub-picture sequence is typically stored in its own track. There are a few options for the storage of a coded tile or sub-picture sequence as a track, which are summarized in Table 5. A sub-picture track contains a sub-picture bitstream and can be decoded with a regular decoding process of AVC or HEVC. Slice headers of a sub-picture track always indicate the sub-picture to appear in the top-left corner of the picture. A tile track contains only a coded tile sequence with its original slice headers, indicating the tile location where it appeared during the encoding. A bitstream can be reconstructed in the form that it was encoded by combining the content from all its tile tracks. An HEVC tile base track references HEVC tile tracks in their order in the coded picture and, hence, facilitates bitstream reconstruction. However, many viewport-dependent streaming approaches combine tile tracks originating from several bitstreams, which may require rewriting of parameter sets and slice headers.

C. Tile Binding

OMAF supports both author-driven and late tile binding approaches. In author-driven tile binding, the processing that requires knowledge of the video coding format is performed by content authors and OMAF players follow instructions created as a part of the content authoring process to merge tiles. In late tile binding, OMAF players rewrite high-level syntax structures of a video bitstream to merge tiles. Both tile binding approaches are described in further detail in the following.

In author-driven tile binding, an extractor track contains instructions to extract data from other tracks and is resolved into a single video bitstream. Extractor tracks are specified in the ISOBMFF encapsulation format of HEVC and AVC bitstreams (ISO/IEC 14496-15). In author-driven tile binding, an extractor track serves as a prescription for OMAF players how tiles are merged from other tracks. An extractor track also contains rewritten parameter sets and slice headers since they cannot typically be inherited from the referenced tracks.

In free-viewport author-driven tile binding, an extractor track suits any viewing orientation (hence, the qualifier free-viewport) and provides multiple options for how tiles can be merged. For example, an extractor track may contain references to track groups, each containing collocated tiles of different bitrates. An OMAF player can choose tiles

Table 5 Storage Options for Coded sub-picture and Tile Sequences

Sample entry	Description
`avc1`, `avc3`	AVC sub-picture track
`hvc1`	HEVC sub-picture track
`hvt1`	HEVC tile track
`hvt2`	HEVC slice segment data track, which does not contain slice headers and must be referenced by an extractor track for bitstream reconstruction
`hvt3`	HEVC tile track, which contains slice header information in-band to simplify late tile binding

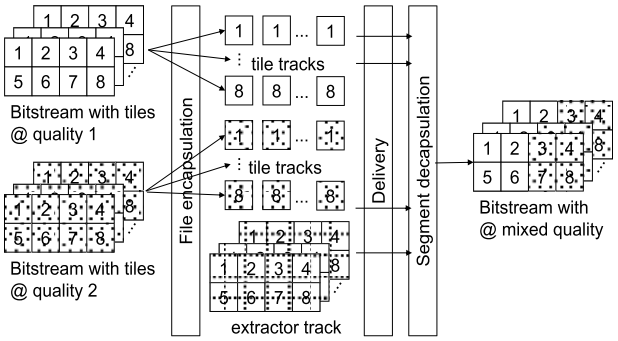

Fig. 9. *Example of free-viewport author-driven tile binding.*

covering the viewport so that they have higher bitrate and/or picture quality than the tiles selected for the other parts of the sphere.

Content authoring for free-viewport author-driven tile binding is illustrated through an example in Fig. 9. ERP content is encoded with 4×2 tiles at two qualities. Each encoded tile sequence is stored as a tile track. Each pair of collocated tile tracks may be encapsulated into the same track group. An extractor track is also created, where each tile location may reference the track group of that location, thus indicating that a player should choose which of the two tile tracks is received for that location. The figure illustrates one possible player's selection for the tile tracks to be received and merged into a bitstream with tiles of mixed quality.

In VS author-driven tile binding, each extractor track is tailor-made for a certain range of viewing orientations, described by RWQR metadata. Thus, the content author must prepare several extractor tracks to cover all possible viewing orientations. An OMAF player selects an extractor track based on its RWQR metadata so that the viewport is covered by higher quality than the remaining parts of the sphere.

Fig. 10 presents an example of content authoring for VS author-driven tile binding, where CMP content is encoded at two resolutions, with 2×2 tiles per cube face. While not presented in the figure, each encoded tile sequence is stored as a tile track. Moreover, several extractor tracks are created by selecting 12 high-resolution tiles covering a

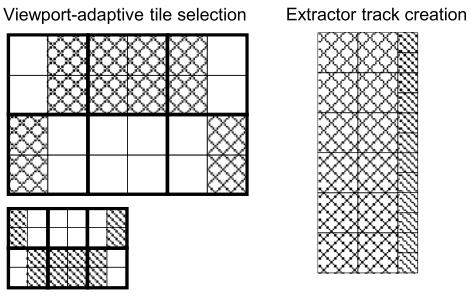

Fig. 10. *Example of content authoring for the VS author-driven tile binding.*

certain range of viewing orientations and the remaining tiles from the low-resolution CMP. Each sample in an extractor track hence contains instructions to copy slice data from selected tile tracks. The figure illustrates one possible selection of the tiles in relation to the CMP format and the spatial arrangement according to which the extractor track organizes the tiles into a coded picture. The RWP metadata of the extractor track describe the mapping between rectangular regions in the decoded pictures and the CMP picture format.

In late tile binding, an OMAF player selects the tiles to be received and merges them into a single video bitstream. Late tile binding gives freedom to OMAF players, e.g., on selecting the field of view for the viewport but also requires more sophisticated client-side processing compared to author-driven tile binding.

An OMAF base track provides instructions to reconstruct a single video bitstream by merging samples of the referenced tile or sub-picture tracks. An OMAF base track can either be an HEVC tile base track or an extractor track. When late tile binding is targeted, the OMAF base track is typically an HEVC tile base track due to its low byte count overhead. However, it is remarked that, even if extractor tracks were provided by the content author, an OMAF player could choose to ignore them and perform late tile binding.

Several versions of the content at different resolutions and possibly for different bitrates or different random access point periods are encoded. The tile tracks that have the same resolution and are collocated may be encapsulated into the same track group to indicate that they are alternatives out of which players should choose at most one track. The same tile dimensions are typically used across all resolution versions to simplify the merging of tile tracks in any order.

In late tile binding, an OMAF player performs the following operations for bitstream rewriting.

1) The parameter sets in the initialization segment in the main adaptation set can be used as the basis but need to be modified according to the selected tile adaptation sets.
2) The spatial location of a slice in the merged bitstream may differ from its location in the encoded bitstream, and when it does differ, rewriting of the slice header is needed.
3) Removal and insertion of the start code emulation prevention bytes may be needed depending on the rewritten syntax structures of parameter sets and slice headers.

An example of late tile binding is illustrated in Fig. 11. CMP content is encoded at two resolutions (2048×2048 and 512×512 per cube face) and the same tile size (512×512). Each encoded tile sequence is stored as a tile track, out of which an OMAF player can select any set of tile tracks to be received. The coded slices are decapsulated from the received tile tracks, and their slice

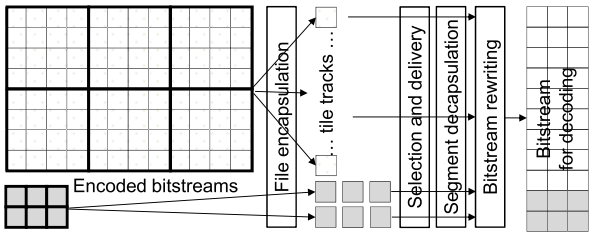

Fig. 11. ***Example of late tile binding.***

headers are rewritten so that a conforming video bitstream is obtained. In this example, the OMAF player selects all low-resolution tile tracks as a fallback to cope with sudden viewing orientation changes and 27 high-resolution tile tracks covering the viewport.

D. Tile Index and Tile Data Segment Formats

In tile-based viewport-dependent 360° streaming, the number of representations can be relatively high, even up to hundreds of Representations, since the content may be partitioned into several tens of tiles and maybe coded with several resolutions and bitrates. Moreover, the duration of (sub)segments may be inconveniently long to update the viewport quickly with high-quality tiles after a viewing orientation change. Thus, requests having a finer granularity than (sub)segments could be desirable. To enable fine-grained requests, even down to a single picture interval, and to obtain the indexing data conveniently for all tiles, OMAF v2 includes new segment formats, namely, initialization segment for an OMAF base track, a tile index segment, and a tile data segment.

The initialization segment for an OMAF base track contains the track header for the OMAF base track and all the referenced tile or sub-picture tracks. This allows the client to download only the initialization segment for the OMAF base track without the need to download the initialization segments of the referenced tile or sub-picture tracks.

The tile index segment is logically an index segment as specified in the DASH standard. It is required to include `MovieFragmentBox`es for the OMAF base track and all the referenced tile or sub-picture tracks. `MovieFragmentBox`es indicate the byte ranges on a sample basis. Consequently, a client can choose to request content on smaller units than (sub)segments.

The tile data segments are media segments containing only media data enclosed in `IdentifiedMediaDataBox`es (“`imda`”). The byte offsets contained in `MovieFragmentBox`es (“`moof`”) are relative to the start of `IdentifiedMediaDataBox`es. Thus, `MovieFragmentBox`es and media data can reside in separate resources, unlike in conventional DASH segment formats where the byte offsets to the media data are relative to the `MovieFragmentBox`. The box payload of each `IdentifiedMediaDataBox` starts with a sequence number that is also contained in the corresponding `MovieFragmentBox`, thus enabling to pair a `MovieFragmentBox` with the corresponding `IdentifiedMediaDataBox`.

E. Content Authoring

Since OMAF supports many types of viewport-dependent streaming, a content author has the freedom to choose which approach is used for preparing the content. Thus, the viewport-dependent streaming approach needs to be selected first. Preparation of multiple VS 360° streams would require preprocessing (e.g., generation of regionwise mixed-resolution content), spatially tailored encoding, and/or rewriting of encoded streams. The choice between tile-based viewport-dependent streaming approaches may depend on the resolution of the original content, the expected decoding capability, and the expected display resolutions. The targeted OMAF video profile also limits the choice that codecs and viewport-dependent streaming approaches can be supported, as indicated in Table 6.

A benefit of both the viewport + 360° video and RWMR methods is that they enable improving the resolution on the viewport with a constrained video decoding capacity. For example, in [27], it was shown that the viewport can originate from a 6K (6144 × 3072) version of the content even though the decoding capacity of the OMAF player only ranges up to about 4K (4096 × 2048) resolution. This article also compared the rate-distortion performance of RWMR and RWMQ approaches. An advantage of RWMR compared to the viewport + 360° technique is that no decoding capacity is spent for decoding low-resolution video that is superimposed by the high-resolution tiles.

Some devices may have problems downloading tens of HTTP streams in parallel, each requiring bandwidth of up to several Mb/s. It is, therefore, advisable to keep the number of required tile or sub-picture representations for the author-driven tile binding at the lower end of the range allowed by the codec at least in some extractor or tile base tracks.

In the following, we concentrate on the tile-based operation of HEVC, while an AVC-based pipeline could be implemented similarly. The content authoring workflow for tile-based viewport-dependent operation is depicted in Fig. 12, and the steps of the workflow are described in the next paragraphs. For practical implementation examples, the Nokia OMAF reference implementation [4] covers steps 2–6 described below, and HEVC encoding with tiles is supported for example in the HM reference software [36] and in the Kvazaar open-source software [37].

1) *Encoding:* The video content is encoded using tiles or the content is split into sub-picture sequences before encoding and then encoded in a constrained manner so that merging of the coded sub-picture sequences into the same bitstream is possible. Usually, multiple versions of the content are generated at different bitrates. A relatively short random access interval,

Table 6 OMAF Video Profiles

OMAF video profile	OMAF version	Codec	Bit depth	Decoding capacity	Projection formats	Viewport-dep. streaming	Tile/sub-picture tracks	Tile binding	Segment formats
HEVC-based viewport-independent	v1	HEVC Main 10	≤10 bits	4K @ 60 Hz	ERP	-	-	-	Conventional
Unconstrained HEVC-based viewport-independent	v2	HEVC Main 10	≤10 bits	Not constrained	ERP	-	-	-	Conventional
HEVC-based viewport-dependent	v1	HEVC Main 10	≤10 bits	4K @ 60 Hz	ERP, CMP	VS streams, tile-based	`hvc1`	Author-driven, late (optional)	Conventional
AVC-based viewport dependent	v1	AVC Progressive High	8 bit	4K @ 30 Hz	ERP, CMP	VS streams, tile-based	`avc1`, `avc3`	Author-driven, late (optional)	Conventional
Simple tiling	v2	HEVC Main 10	≤10 bits	Not constrained	ERP, CMP	Tile-based	`hvc1`, `hvt1`, `hvt2`, `hvt3`	Author-driven, late (optional)	Tile Index and Tile Data
Advanced tiling	v2	HEVC Main 10	≤10 bits	Not constrained	3D mesh	Tile-based	`hvt3`	Late	Tile Index and Tile Data

e.g., in the order of 1 s, is used in encoding to enable frequent viewport switching.

2) *Bitstream Processing*: A processing step may be needed to prepare the encoded bitstreams for encapsulation into sub-picture or tile tracks. When the content was encoded using tiles, each tile sequence is extracted from the bitstream. This requires parsing of the high-level structure of the bitstream, including parameter sets and slice headers. When sub-picture bitstreams were encoded, no additional processing at this phase is needed.
3) *sub-picture or Tile Track Generation:* OMAF video profiles constrain that sample entry types are allowed for the sub-picture or tile tracks. Slice headers require rewriting in all cases where the slice position in the encoded bitstream does not match the position implied by the sample entry type. As an integral part of generating both the sub-picture or tile tracks and the extractor or tile base track(s), the necessary OMAF file format metadata is also authored.
4) *Extractor or Tile Base Track Generation:* If the "`hvt1`" or "`hvt3`" sample entry type is in use, a tile base track is generated. Otherwise, one or more extractor tracks are created. A single extractor track is typically sufficient for free-viewport author-driven tile binding, whereas one extractor track per a distinct viewing direction may be needed for VS author-driven tile binding.
5) *(Sub)segment Encapsulation:* (Sub)segments are created from each track for DASH delivery. When conventional segment formats specified in the DASH standard are in use, no changes to the (sub)segment encapsulation process are needed compared to the corresponding process for 2-D video.
6) *DASH MPD Generation:* An MPD is generated. Each extractor track and tile base track form a representation in its own adaptation set. An adaptation set consists of the sub-picture or the tile representations covering the same sphere region at the same resolution but at different bitrates. The DASH preselection feature is used to associate the extractor or tile base adaptation set with the associated sub-picture or tile adaptation sets. Moreover, in this processing step, the OMAF file metadata is interpreted to create the OMAF extensions for DASH MPD.

V. OMAF VIDEO PROFILES

A summary of the video profiles specified in OMAF is presented in Table 6. This section first introduces the video profiles and then discusses the similarities and differences between the profiles.

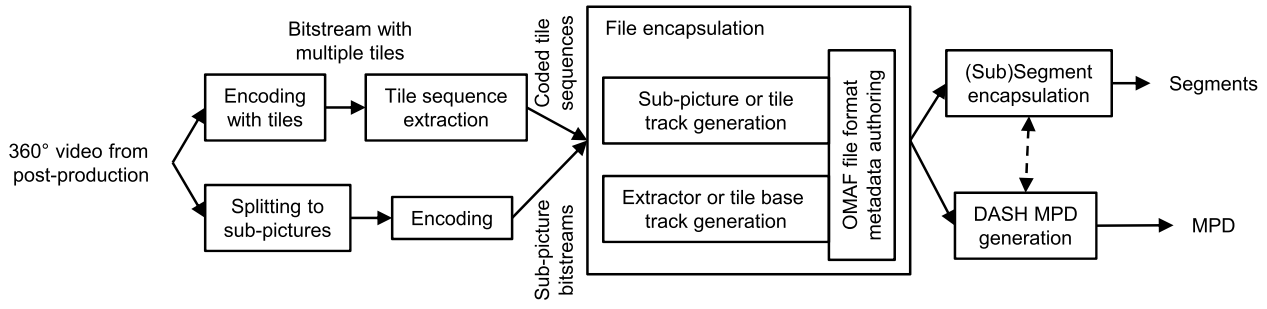

Fig. 12. ***Basic flow of content authoring operations for tile-based viewport-dependent streaming.***

The HEVC-based viewport-independent profile is intended for basic viewport-independent files and streaming using the ERP. In OMAF v1, the decoding capacity of the HEVC-based viewport-independent profile was limited to approximately 4K (4096 × 2048) resolution at 60-Hz picture rate, while the unconstrained HEVC-based viewport-independent was specified similarly in OMAF v2 but without decoding capacity constraints to respond to the need of higher HMD resolutions and the availability of more powerful video decoding hardware.

The HEVC- and AVC-based viewport-dependent profiles support both VS streaming and different types of tile-based viewport-dependent streaming schemes. Two tiling profiles, namely the simple and advanced tiling profiles, were added for viewport-dependent streaming in OMAF v2. The main difference of the simple tiling profile compared to the HEVC-based viewport-dependent profile is the use of the tile index and tile data segment formats. The advanced tiling profile is the only profile that uses the 3-D mesh projection format and requires players to support late tile binding, while, otherwise, it is similar to the simple tiling profile.

Bit Depth: Since the HEVC-based profiles require support for the HEVC Main 10 Profile, they support bit depths up to 10 bits, whereas the AVC-based viewport-dependent profile is limited to 8 bits per color component.

Decoding Capacity: The HEVC-based profiles specified in OMAF v1 require support for Level 5.1, which, in practice, means decoding capacity of approximately 4K pictures at 60 Hz, whereas the AVC-based profile can support only 4K pictures at 30 Hz. The profiles specified in OMAF v2 are tailorable in terms of decoding capacity, and thus, no HEVC level constraints are specified for them.

Projection Formats and RWP: In the HEVC-based viewport-independent profile, the use of RWP can only be used to indicate a limited content coverage. In the HEVC- and AVC-based viewport-dependent profiles, RWP is not constrained. In the simple tiling profile, RWP is, otherwise, unconstrained, but a single region is not allowed to cross a boundary of a projection surface, such as a cube face boundary. Moreover, the RWP format of an OMAF base track is not indicated but inherited by OMAF players from the selected tile or sub-picture tracks. Consequently, OMAF base tracks can enable free-viewport author-driven tile binding. In the advanced tiling profile, the 3-D mesh format is used, and RWP is disabled.

Viewport-Dependent Streaming: The HEVC- and AVC-based viewport-dependent profiles enable both VS streams and tile-based viewport-dependent streaming, while the simple and advanced tiling profiles only enable the latter. While both the HEVC- and AVC-based viewport-dependent profiles support all categories, the AVC-based profile is more constrained since AVC does not support tile partitioning, arranging slices vertically imposes restrictions on slice sizes, and AVC has limits on picture aspect ratio. The advanced tiling profile requires using HEVC tiles of identical width and height and a tile track to contain exactly one HEVC tile. Compared to the advanced tiling profile, the HEVC-based viewport-dependent and the simple tiling profiles provide more freedom since they enable using rectangular slices that comprise one or more tiles or a subset of a tile as the unit for tile-based streaming.

VI. OMAF IMAGE PROFILES

The image profiles of OMAF were designed to be seamlessly compatible with HEIF. Consequently, devices and platforms with HEIF capability are easily extensible to support 360° images with metadata specified in OMAF. Since OMAF is a toolbox standard, it is envisioned that devices could only implement specific parts of OMAF. For example, 360° cameras could only support an OMAF image profile or the HEIF image metadata specified in the OMAF standard.

At the time of releasing OMAF v1, there was arguably no other standard for storage of 360° images with the necessary metadata for displaying them properly. Since then, the JPEG 360 standard [33] was finalized and includes omnidirectional metadata specifications for JPEG [34] and JPEG 2000 [35] images. Since OMAF specifies the omnidirectional image metadata for HEIF files, there is no overlap with JPEG 360 even though the types of metadata in OMAF and JPEG 360 are similar.

OMAF v2 integrates images more tightly to 360° presentations that can contain timed media types too. Images can be used as overlays enriching an omnidirectional video background. An opposite arrangement is equally supported, i.e., an omnidirectional background image can be accompanied by video overlays. Moreover, presentations with multiple viewpoints can equally use images or video clips as the visual content of the viewpoints.

OMAF v1 specifies two profiles for projected omnidirectional images. OMAF HEVC image profile uses the HEVC Main 10 profile and the OMAF legacy image profile using the JPEG codec, as summarized in Table 7. Both OMAF image profiles are compatible with HEIF, and they share common features, as listed in Table 8. Coded image items of the OMAF HEVC image profile are limited to approximately the 4K resolution, but larger image sizes can be achieved by using the "`grid`" derived image item, which arranges input images onto a grid to create a large output image. The image resolution constraint ensures that most

Table 7 OMAF Image Profiles

OMAF image profile	Codec	Bit depth
OMAF HEVC image profile	HEVC Main 10 Profile	≤10 bits
OMAF legacy image profile	JPEG	8 bit

Table 8 Features of OMAF Image Profiles

Projection formats	ERP, CMP
Mono/stereo	Monoscopic and frame-packed strereoscopic
Region-wise packing	For ERP: Up to one packed region per view For CMP: Up to one packed region per cube face
Content coverage	≤ 360°

Table 9 OMAF Toolset Brands

Brand name	4CC	Main usage
Viewpoint	vwpt	Freely navigable multi-camera presentation
Non-linear storyline	nlsl	Presentation with user-selectable storyline paths
Overlay	ovly	Sphere- or viewport-relative video or image overlays

hardware implementations can be used for HEVC image decoding.

VII. OMAF TOOLSET BRANDS

A. Introduction

OMAF v2 specifies viewpoint, nonlinear storyline, and overlay toolset brands, which are summarized in Table 9. Compatibility to a toolset brand can be indicated at the file level using the 4CC of the brand. This section reviews the OMAF features for multiple viewpoints and overlays, as well as the toolset brands.

B. Multiple Viewpoints

OMAF v2 supports 360° video content comprising pieces captured by multiple 360° video cameras or camera rigs, referred to as viewpoints. This way, users can switch between different viewpoints, e.g., in a basketball game switch between scenes captured by 360° video cameras located at different ends of the court.

Switching between viewpoints captured by 360° video cameras that can "see" each other can be seamless in the sense that after switching the user still sees the same object, e.g., the same player in a sports game, just from a different viewing angle. However, when there is an obstacle, e.g., a wall, between two 360° video cameras such that they cannot "see" each other, switching between the two viewpoints incurs a noticeable cut or transition.

When multiple viewpoints exist, identification and association of tracks or image items belonging to one viewpoint are needed. For this purpose, OMAF specifies the viewpoint grouping of tracks and image items, as well as similar metadata for DASH MPD. This grouping mechanism provides an identifier (ID) of the viewpoint and a set of other information that can be used to assist streaming of the content and switching between different viewpoints. Such information includes the following.

1) A label, for annotation of the viewpoint, e.g., "home court."
2) Mapping of the viewpoint to a viewpoint group consisting of cameras that "see" each other and have an indicated viewpoint group ID. This information provides a means to indicate whether the switching between two particular viewpoints can be seamless.
3) Viewpoint position relative to the common reference coordinate system shared by all viewpoints of a viewpoint group. Viewpoint positions enable a good user experience during viewpoint switching, provided that the client can properly utilize the positions in its rendering process.
4) Rotation information for conversion from the global coordinate system of the viewpoint to the common reference coordinate system.
5) Optionally, the orientation of the common reference coordinate system relative to the geomagnetic north.
6) Optionally, the global positioning system (GPS) location of the viewpoint, which enables the client application to place the viewpoint into a real-world map.
7) Optionally, viewpoint switching information, which provides a number of switching transitions possible from the current viewpoint, and for each of these, information such as the sphere region that a user can select to cause the viewpoint switch, the destination viewpoint, the viewport to view after switching, the presentation time to start the playback of the destination viewpoint, and a recommended transition effect during switching (such as zoom-in, walk though, fade-to-black, or mirroring).
8) Optionally, viewpoint looping information indicating which time period of the presentation is looped and a maximum count of how many times the time period is looped. The looping feature can be used for requesting end-user's input for initiating viewpoint switching.

Some of the viewpoints can be static, i.e., captured by 360° video cameras at fixed positions. Other viewpoints can be dynamic, e.g., captured by a 360° video camera mounted on a flying drone. For dynamic viewpoints, the above information is stored in timed metadata tracks that are time-synchronized with the media tracks.

C. Nonlinear Storyline

The viewpoint switching and looping information enable content authors to generate presentations with a nonlinear storyline. Each viewpoint is a scene in the storyline. The viewpoint switching metadata can be used to provide multiple switching options from which an end-user is required to choose before advancing to the next scene of the storyline. The user selection may be linked to a given sphere region, viewport region, or overlay, but other user input means are not precluded either. The viewpoint looping metadata may be used to create a loop in the playback of the current scene to wait for the user's selection. The viewpoint looping metadata also allow defining a default destination viewpoint that is applied when an indicated maximum number of loops has been passed.

Fig. 13 presents an example where Scene 1 is played until the end of its timeline, and then, a given time range of Scene 1 is repeated until an end-user selects between Scenes 2a and 2b. After completing the playback of Scene 2a or 2b, the playback automatically switches to Scene 3, after which the presentation ends.

D. Overlays

An overlay is a video clip, an image, or text that is superimposed on top of an omnidirectional video or image.

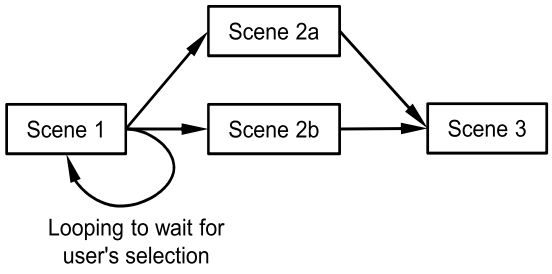

Fig. 13. *Nonlinear storyline example.*

Overlays can be used for multiple purposes, including the following:

1) annotations of the content; for instance, stock tickers and player statistics of sports games;
2) recommended viewport for the content, for example, giving the end-user the possibility to follow the director's intent while having the freedom to peek freely around;
3) 2-D video or image close-ups of the omnidirectional video or image on the background;
4) hotspots for switching viewpoints interactively;
5) displaying a logo of the content provider;
6) displaying a semitransparent watermark on top of the content;
7) advertisements.

The appearance of overlays can be controlled flexibly in OMAF. Moreover, the overlay structures are extensible, and new controls or properties can be specified in future versions or amendments of the OMAF standard. Some basic concepts related to overlays are illustrated in Fig. 14, which shows an equator-level cross section of the unit sphere and different types of overlays. Background visual media is defined as the omnidirectional video or image that is rendered on the unit sphere, and the term overlay source refers to the visual content displayed as an overlay.

The following types of overlays are specified in OMAF.

1) Sphere-relative 2-D overlays, where an overlay source is displayed on a plane of a given width and height. The center point of the plane is located at given spherical coordinates and distance from the center of the unit sphere, and the plane can be rotated by given yaw, pitch, and roll angles.
2) Sphere-relative omnidirectional overlays, where an omnidirectional projection, such as ERP, has been used for an overlay source. Sphere-relative omnidirectional overlays may, but need not, cover the entire sphere and are located at a given spherical location and distance from the center of the unit sphere.
3) Sphere-relative 3-D mesh overlays, where both a 3-D mesh and a mapping of an overlay source onto the 3-D mesh are specified. The 3-D mesh can consist of parallelograms having any rotation and being located at any position within the unit sphere.
4) Viewport-relative overlays, which are located on a given position within the viewport regardless of the viewing orientation. The rendering process projects the sphere-relative overlays and the background visual media onto the viewport, which is then superimposed by the viewport-relative overlays. This is illustrated in Fig. 14 through an isosceles triangle whose sides illustrate the horizontal field of view of a display and the base corresponds to a viewport. Since viewports can be of different shapes and sizes in different player devices, the top-left corner position, width, and height of a viewport-relative overlay are provided in percents relative to the viewport dimensions.

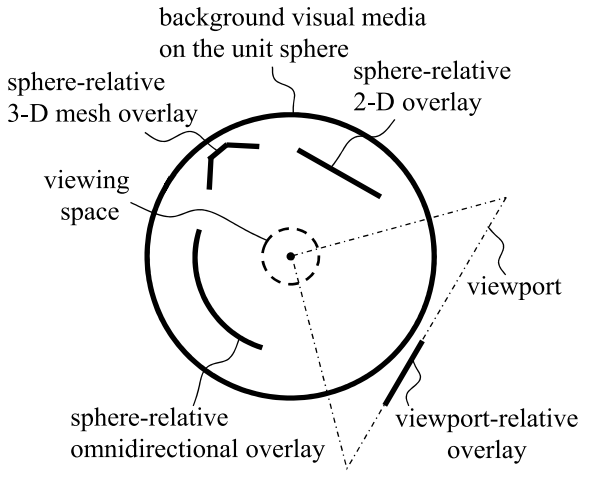

Fig. 14. *2-D illustration of overlays and background visual media.*

OMAF enables two rendering modes for presenting sphere-relative overlays with background visual media. In conventional 3DOF rendering, a viewing position that is in the center of the unit sphere is used for projecting the sphere-relative overlays and the background visual media onto the viewport. In the second rendering mode, the viewing position is tracked and used for projecting the content onto the viewport. When the second rendering mode is used with an HMD, it may be referred to as head-tracked rendering. The second rendering mode enables viewing overlays from different perspectives and peeking on the background appearing behind the overlays. Sphere-relative overlays can be placed at given distances from the center of the unit sphere, which is perceivable through motion parallax. Content authors can define a viewing space that specifies valid viewing positions around the center of the unit sphere. OMAF enables specifying the viewing space boundaries as a cuboid, a sphere, a cylinder, or an ellipsoid.

As discussed above, sphere-relative overlays are located at a given distance from the center of the unit sphere. A layering order can be given so that the player behavior is deterministic when several overlays are positioned at the same distance or when viewport-relative overlays overlap.

By default, overlays are opaque. However, either a constant opacity or an alpha plane that specifies a pixelwise opacity can be optionally provided.

The content author can specify, separately per each overlay, which types of user interactions are enabled. The following user interaction types can be enabled or disabled in an OMAF file: changing the position, modifying the distance from the center of the unit sphere, switching the overlay ON or OFF, tuning the opacity, resizing, rotating, cropping, and switching the overlay source to another one. A textual label can be given for each overlay and utilized by a user interface to enable end-users to switch overlays ON or OFF. Another way is to provide an associated sphere region that the user can select to turn an overlay ON or OFF.

As discussed above, an overlay source can either be a video track or an image item; in that case, the overlay consists of the entire decoded picture. Since some player devices might not be capable of running several video decoder instances simultaneously, it is also possible to pack overlays spatially with the background visual media. In that case, an overlay source is specified as a rectangle within the decoded picture area. Furthermore, it is possible to indicate that an overlay source is defined by the recommended viewport timed metadata track or provided by external means, such as through a URL. The externally specified overlay source could be used to show content from a separate application within an OMAF presentation.

The content author has two mechanisms to enable scaling the player-side complexity of overlay rendering. First, each overlay can be given a priority for rendering. The highest priority value means that the overlay must be rendered. Second, it is indicated whether control or property associated with an overlay is essential or optional. For example, it can be indicated that overlay composition with an alpha plane is optional. In this case, if the player does not have enough resources to carry out the processing required for alpha planes, it is allowed to render an opaque overlay.

The controls and properties for overlays can be static, i.e., remain constant for the entire duration of the overlay, or dynamic, i.e., signaled by a timed metadata track where the controls and properties are dynamically adjusted. For example, it is possible to move or resize an overlay as a function of time.

VIII. CONCLUSION

An overview of OMAF, the arguably first VR system standard, was provided. The overview focused on omnidirectional video and images, without much detail on audio and timed text. This article described the OMAF architecture, the representation formats for omnidirectional video and images, and the file format and DASH extensions. Furthermore, 360° video streaming techniques and the related features in OMAF were discussed in detail. In addition, the OMAF video and image profiles, as well as the toolsets for overlays, viewpoints, and nonlinear storylines, were summarized.

The OMAF standard supports many different approaches for viewport-dependent streaming. It is an open research question which approach provides the best end-user experience. Furthermore, there are many detailed research topics that would benefit from a more thorough investigation, for example, determination of optimal projection format or 3-D mesh, tiling strategy, and bitrate adaptation logic for tile-based streaming.

Requirements for the next OMAF version have been agreed in MPEG [38] and include support for new visual volumetric media types, namely, video-based point cloud compression (V-PCC) and immersive video. The MPEG standard for visual volumetric video-based coding and V-PCC [39] was recently finalized and can be used to represent captured volumetric objects. The MPEG Immersive Video standard [40] has a target completion by July 2021 and enables 6DOF within a limited viewing volume. It is expected that the OMAF standardization for integrating these media types will start in 2021.

Acknowledgment

The authors would like to greatly thank the numerous Moving Picture Experts Group (MPEG) delegates who have contributed to the development of Omnidirectional MediA Format (OMAF). They also express gratitude to the coeditors with whom the authors had a pleasure to work either in v1 or v2 of OMAF. They are also grateful to the anonymous reviewers and Lukasz Kondrad for their excellent suggestions to improve this article. ■

REFERENCES

[1] R. S. Kalawsky, *The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments*. Reading, MA, USA: Addison-Wesley, 1993.

[2] F. Biocca and M. R. Levy, Eds., *Communication in the Age of Virtual Reality*. Newark, NJ, USA: Lawrence Erlbaum Associates, 1995.

[3] *Information Technology—Coded Representation of Immersive Media—Part 2: Omnidirectional Media Format*, Standard ISO/IEC 23090-2:2019, 2019.

[4] *Nokia OMAF Implementation*. Accessed: Mar. 9, 2021. [Online]. Available: https://github.com/nokiatech/omaf

[5] D. Podborski *et al.*, "HTML5 MSE playback of MPEG 360 VR tiled streaming: JavaScript implementation of MPEG-OMAF viewport-dependent video profile with HEVC tiles," in *Proc. 10th ACM Multimedia Syst. Conf.*, Jun. 2019, pp. 324–327. [Online]. Available: https://www.youtube.com/watch?v=FpQiF8YEfY4 and https://github.com/fraunhoferhhi/omaf.js

[6] *Open Visual Cloud Immersive Video Samples*. Accessed: Mar. 9, 2021. [Online]. Available: https://github.com/OpenVisualCloud/Immersive-Video-Sample

[7] S. Deshpande, Y.-K. Wang, and M. M. Hannuksela, Eds., *Text of ISO/IEC FDIS 23090-2 2nd edition OMAF*, document ISO/IEC JTC1 SC29 WG3, N00072, Dec. 2020.

[8] K. K. Sreedhar, I. D. D. Curcio, A. Hourunranta, and M. Lepistö, "Immersive media experience with MPEG OMAF multi-viewpoints and overlays," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 333–336. [Online]. Available: https://www.youtube.com/watch?v=WcucAw3HNVE

[9] *How ClearVR Drives and Leverages Standards*. Accessed: Oct. 27, 2020. [Online]. Available: https://www.tiledmedia.com/index.php/standards/

[10] *VR Industry Forum Guidelines, Version 2.3*. Accessed: Jan. 2021. [Online]. Available: https://www.vr-if.org/guidelines/

[11] *VR Industry Forum Newsletter*. Accessed: Dec. 2020. [Online]. Available: https://www.vr-if.org/december-2020-newsletter/

[12] *Virtual Reality (VR) Profiles for Streaming Applications*, document 3GPP Technical Specification 26.118, 2020. Accessed: Oct. 27, 2020. [Online]. Available: https://www.3gpp.org/ftp//Specs/archive/26_series/26.118/

[13] *Information Technology—Coding of Audio-Visual Objects—Part 12: ISO Base Media File Format*, Standard ISO/IEC 14496-12, 2012.

[14] *Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, Standard ISO/IEC 23009-1:2019, 2019.

[15] M. M. Hannuksela, Y.-K. Wang, and A. Hourunranta, "An overview of the OMAF standard for 360° video," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, pp. 418–427.

[16] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 12: Image File Format*, Standard ISO/IEC 23008-12, 2012.

[17] M. M. Hannuksela, E. B. Aksu, V. K. M. Vadakital, and J. Lainema, *Overview of the High Efficiency Image File Format*, document JCTVC-V0072, Oct. 2015, pp. 1–12. [Online]. Available: http://phenix.it-sudparis.eu/jct/doc_end_user/documents/22_Geneva/wg11/JCTVC-V0072-v1.zip

[18] *Advanced Video Coding*, document ITU-T Rec. H.264, ISO/IEC 14496-10, 2010.

[19] *High Efficiency Video Coding*, document ITU-T Rec. H.265, ISO/IEC 23008-2, 2002.

[20] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, "360 degrees video coding using region adaptive smoothing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 750–754.

[21] R. G. Youvalari, A. Aminlou, and M. M. Hannuksela, "Analysis of regional down-sampling methods for coding of omnidirectional video," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2016, pp. 1–5.

[22] M. Tang, Y. Zhang, J. Wen, and S. Yang, "Optimized video coding for omnidirectional videos," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 799–804.

[23] Y. Li, J. Xu, and Z. Chen, "Spherical domain rate-distortion optimization for omnidirectional video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1767–1780, Jun. 2019.

[24] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proc. ACM Multimedia Conf.*, Oct. 2016, pp. 601–605.

[25] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2016, pp. 583–586.

[26] R. Ghaznavi-Youvalari *et al.*, "Comparison of HEVC coding schemes for tile-based viewport-adaptive streaming of omnidirectional video," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[27] A. Zare, A. Aminlou, and M. M. Hannuksela, "6K effective resolution with 4K HEVC decoding capability for OMAF-compliant 360° video streaming," in *Proc. 23rd Packet Video Workshop*, Jun. 2018, pp. 72–77.

[28] H. Hristova, X. Corbillon, G. Simon, V. Swaminathan, and A. Devlic, "Heterogeneous spatial quality for omnidirectional video," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–6.

[29] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," *Proc. IEEE*, vol. 93, no. 1, pp. 98–110, Jan. 2005.

[30] P. R. Alface, J.-F. Macq, and N. Verzijp, "Evaluation of bandwidth performance for interactive spherical video," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011, pp. 1–6.

[31] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 259–267, Apr. 2004.

[32] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, "Tile based HEVC video for head mounted displays," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 399–400.

[33] *Information Technology—JPEG Systems—JPEG 360*, Standard ISO/IEC 19566-6:2019, 2019.

[34] *Digital Compression and Coding of Continuous-Tone Still Images*, Standard ISO/IEC 10918-1:1994, 1994.

[35] *JPEG 2000 Image Coding System*, Standard ISO/IEC 15444-1:2019, 2019.

[36] *Reference Software for High Efficiency Video Coding*, document ITU-T Rec. H.265.2, Dec. 2016, ISO/IEC 23008-5:2017, 2017.

[37] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, "Kvazaar 2.0: Fast and efficient open-source HEVC inter encoder," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 237–242. [Online]. Available: https://github.com/ultravideo/kvazaar

[38] M.-L. Champel and I. D. D. Curcio, Eds., *Requirements for MPEG-I Phase 2*, document ISO/IEC JTC1 SC29 WG11, N19511, Jul. 2020.

[39] *Visual Volumetric Video-Based Coding and Video-Based Point Cloud Compression*, document ISO/IEC JTC1 SC29 WG11, N19579, Sep. 2020.

[40] *MPEG Immersive Video*, document ISO/IEC CD 23090-12, ISO/IEC JTC1 SC29 WG11, N19482, Jul. 2020.

ABOUT THE AUTHORS

Miska M. Hannuksela (Member, IEEE) received the M.Sc. degree in engineering and the D.Sc. degree in technology from the Tampere University of Technology, Tampere, Finland, in 1997 and 2010, respectively.

He has been with Nokia Technologies, Tampere, since 1996, in different roles including research manager/leader positions in the areas of video and image compression, end-to-end multimedia systems, and sensor signal processing and context extraction. He is currently the Bell Labs Fellow and the Head of Video Research, Nokia Technologies. He has published above 180 journal articles and conference papers and more than 1000 standardization contributions in Joint Video Experts Team (JVET), Joint Collaborative Team on Video Coding (JCT-VC), Joint Video Team (JVT), Moving Picture Experts Group (MPEG), the 3rd Generation Partnership Project (3GPP), and Digital Video Broadcasting Project (DVB). He has granted patents from more than 130 patent families. His research interests include video compression, multimedia communication systems and formats, user experience and human perception of multimedia, and sensor signal processing.

Dr. Hannuksela has several best paper awards and received an award of the best doctoral thesis of the Tampere University of Technology in 2009 and the Scientific Achievement Award nominated by the Centre of Excellence of Signal Processing, Tampere University of Technology, in 2010. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2015. He has been an Editor in several video and systems standards, including the High Efficiency Image File Format (HEIF), the Omnidirectional Media Format, RFC 3984, and RFC 7798 and some parts of H.264/AVC, H.265/High Efficiency Video Coding (HEVC), and the ISO Base Media File Format.

Ye-Kui Wang received the B.S. degree in industrial automation from the Beijing Institute of Technology, Beijing, China, in 1995, and the Ph.D. degree in information and telecommunication engineering from the Graduate School in Beijing, University of Science and Technology of China, Hefei, China, in 2001.

His earlier working experiences and titles include the Chief Scientist of Media Coding and Systems at Huawei Technologies, San Diego, CA, USA, the Director of Technical Standards at Qualcomm, San Diego, CA, a Principal Member of Research Staff at Nokia Corporation, Tampere, Finland, and so on. He is currently a Principal Scientist with Bytedance Inc., San Diego. He has been an active contributor to various multimedia standards, including video codecs, file formats, real-time transport protocol (RTP) payload formats, and multimedia streaming and application systems, developed by various standardization organizations including International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) Video Coding Experts Group (VCEG), ISO/IEC Moving Picture Experts Group (MPEG), Joint Video Team (JVT), Joint Collaborative Team on Video Coding (JCT-VC), Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V), Internet Engineering Task Force (IETF), Audio Video coding Standard (AVS), Digital Video Broadcasting Project (DVB), Advanced Television Systems Committee (ATSC), and Digital Entertainment Content Ecosystem (DECE). He has coauthored about 1000 standardization contributions, over 60 academic articles, and about 500 families of patent applications (out of which 336 U.S. patents have been granted as of February 23, 2021). His research interests include video coding, storage, transport, and multimedia systems.

Dr. Wang has been chairing the development of Omnidirectional MediA Format (OMAF) at MPEG. He has been an Editor for several standards, including versatile video coding (VVC), OMAF, all versions of High Efficiency Video Coding (HEVC), VVC file format, HEVC file format, layered HEVC file format, ITU-T H.271, SVC file format, multiview video coding (MVC), RFC 6184, RFC 6190, RFC 7798, 3GPP TR 26.906, and 3GPP TR 26.948.