
**Information technology — Coded
representation of immersive media —
Part 2:
Omnidirectional media format**

*Technologies de l'information — Représentation codée de média
immersifs —*

Partie 2: Format de média omnidirectionnel





COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword.....	vii
Introduction	viii
1 Scope.....	1
2 Normative references	1
3 Terms, definitions, abbreviated terms and symbols	3
3.1 Terms and definitions	3
3.2 Abbreviated terms	11
3.3 Symbols.....	12
3.3.1 Arithmetic operators and mathematical functions.....	12
3.3.2 Order of operation precedence.....	13
3.3.3 Range notation	14
3.3.4 Variables	14
3.3.5 Processes	14
3.3.6 Syntax structures	15
3.3.7 Conventions for indicating the number of boxes in tables.....	15
4 Overview	15
4.1 Overall architecture	15
4.2 Projected omnidirectional video/images.....	18
4.2.1 General.....	18
4.2.2 Stitching, rotation, projection, and region-wise packing	18
4.3 Fisheye omnidirectional video/images.....	19
4.4 Mesh omnidirectional video.....	20
4.5 Streaming methods for omnidirectional video.....	20
4.5.1 Overview.....	20
4.5.2 Tile-based streaming with viewport-specific author-driven binding.....	22
4.5.3 Tile-based streaming with free-viewport author-driven binding.....	22
4.5.4 Tile-based streaming with late binding	23
4.6 Additional functionalities	25
4.7 Conformance and interoperability	25
4.7.1 General.....	25
4.7.2 Media profiles.....	26
4.7.3 Presentation profiles.....	28
4.7.4 Toolset brands.....	28
4.7.5 Summary of referenceable code points	28
5 Omnidirectional video projection and region-wise packing	35
5.1 Coordinate system.....	35
5.2 Omnidirectional projection formats.....	36
5.2.1 General.....	36
5.2.2 Equirectangular projection for one sample location	36
5.2.3 Cubemap projection for one sample location.....	37
5.3 Conversion from the local coordinate axes to the global coordinate axes	39
5.4 Region-wise packing formats	40
5.4.1 General.....	40
5.4.2 Conversion of one sample location for rectangular region-wise packing	41
6 Fisheye omnidirectional video	42
6.1 General.....	42
6.2 The <code>FisheyeVideoEssentialInfoStruct()</code> syntax structure.....	42
6.2.1 Syntax.....	42
6.2.2 Semantics.....	43
6.3 The <code>FisheyeVideoSupplementalInfoStruct()</code> syntax structure.....	46
6.3.1 Syntax.....	46
6.3.2 Semantics.....	47

7	Omnidirectional media storage and metadata signalling in the ISOBMFF	52
7.1	Generic extensions to the ISOBMFF.....	52
7.1.1	Indication of a track not intended to be presented alone.....	52
7.1.2	Clarifications on the stereo video box.....	52
7.1.3	Generic sub-picture track grouping extensions.....	53
7.1.4	Media offset box.....	57
7.2	Generic extensions to ISO/IEC 14496-15	58
7.2.1	Containing of SpatialRelationship2DDescriptionBox for HEVC tile base track and HEVC tile tracks.....	58
7.3	OMAF-specific extensions to the ISOBMFF	58
7.3.1	Sync samples in timed metadata tracks	58
7.4	OMAF-specific extensions to ISO/IEC 14496-15	59
7.4.1	Coverage information box in an HEVC tile base track.....	59
7.5	Structures and semantics that are common for video tracks and image items	59
7.5.1	Semantics of sample locations within a decoded picture.....	59
7.5.2	Projection format structure	62
7.5.3	Region-wise packing structure.....	63
7.5.4	Rotation structure.....	70
7.5.5	Content coverage structure.....	71
7.5.6	Sphere region structure.....	72
7.6	Restricted video schemes for omnidirectional video	76
7.6.1	Scheme types.....	76
7.6.2	Projected omnidirectional video box.....	81
7.6.3	Fisheye omnidirectional video box.....	82
7.6.4	Region-wise packing box.....	83
7.6.5	Rotation box.....	83
7.6.6	Coverage information box	84
7.6.7	Mesh omnidirectional video box	84
7.6.8	Mesh box.....	85
7.7	Timed metadata for sphere regions	87
7.7.1	General	87
7.7.2	Sample entry.....	88
7.7.3	Sample format.....	89
7.7.4	Initial viewing orientation	89
7.7.5	Recommended viewport.....	91
7.7.6	Timed text sphere location metadata.....	94
7.8	Signalling of region-wise quality ranking	95
7.8.1	General	95
7.8.2	Spherical region-wise quality ranking.....	95
7.8.3	2D region-wise quality ranking.....	97
7.9	Storage of omnidirectional images	99
7.9.1	General	99
7.9.2	Frame packing item property.....	99
7.9.3	Projection format item property	100
7.9.4	Essential fisheye image item property.....	101
7.9.5	Supplemental fisheye image item property.....	102
7.9.6	Region-wise packing item property	102
7.9.7	Rotation item property	103
7.9.8	Coverage information item property.....	103
7.9.9	Initial viewing orientation item property.....	104
7.10	Storage of timed text for omnidirectional video	105
7.10.1	General	105
7.10.2	OMAF timed text configuration box	105
7.10.3	IMSC1 tracks	108
7.10.4	WebVTT tracks	108
7.11	ERP region timed metadata.....	109
7.11.1	General	109
7.11.2	Sample entry format.....	109
7.11.3	Semantics	109
7.11.4	Sample format.....	110

7.11.5	Generating ERP region metadata.....	111
7.12	Storage and signalling of viewpoints for omnidirectional video and images	111
7.12.1	Viewpoint information structures.....	111
7.12.2	Viewpoint entity grouping.....	120
7.12.3	Timed metadata for viewpoints	122
7.13	Storage of omnidirectional video in sub-picture tracks.....	127
7.13.1	General.....	127
7.13.2	Projected omnidirectional video.....	127
7.13.3	Indication of composition pictures being packed pictures or projected pictures	128
7.13.4	Fisheye omnidirectional video.....	128
7.14	Storage and signalling of overlays for omnidirectional video and images.....	129
7.14.1	General.....	129
7.14.2	Overlay structure	131
7.14.3	Overlay control structures.....	132
7.14.4	Overlay configuration box.....	143
7.14.5	Overlay item property	143
7.14.6	Overlay timed metadata track.....	144
7.14.7	Entity groups.....	145
7.14.8	Overlay alpha auxiliary image.....	148
7.15	Signalling of viewing space information.....	149
7.15.1	General.....	149
7.15.2	Viewing space structure.....	149
7.15.3	Viewing space box	152
7.15.4	Viewing space item property.....	152
7.15.5	Time varying immersive viewing space signalling.....	152
7.16	Mapping of rectangular regions to the 3D mesh.....	153
7.16.1	General.....	153
7.16.2	Tile mesh sample grouping	153
7.16.3	Rectangular region structure.....	155
7.16.4	Projection of a sample location onto the 3D mesh.....	156
8	Omnidirectional media encapsulation and signalling in DASH	157
8.1	Architecture of DASH delivery in OMAF	157
8.2	Usage of DASH in OMAF	159
8.2.1	General.....	159
8.2.2	Signalling of stereoscopic frame packing.....	159
8.2.3	Carriage of timed metadata.....	159
8.2.4	Associating Adaptation Sets or Representations with each other	160
8.3	DASH MPD descriptors for omnidirectional media in the namespace "urn:mpeg:mpegI:omaf:2017".....	161
8.3.1	XML namespace and schema	161
8.3.2	Signalling of projection type information	161
8.3.3	Signalling of region-wise packing type.....	162
8.3.4	Signalling of content coverage	163
8.3.5	Signalling of spherical region-wise quality ranking.....	166
8.3.6	Signalling of 2D region-wise quality ranking	172
8.3.7	Signalling of fisheye omnidirectional video.....	177
8.4	Carriage of images	177
8.4.1	General.....	177
8.4.2	Format and constraints for Segments.....	178
8.5	DASH MPD descriptors for omnidirectional media in the namespace "urn:mpeg:mpegI:omaf:2020".....	178
8.5.1	XML namespace and schema	178
8.5.2	Signalling of association.....	178
8.5.3	Signalling of viewpoints.....	180
8.5.4	Signalling of sub-picture composition identifier and its attributes	187
8.5.5	Signalling of overlays	188
8.5.6	Entity to group descriptor	190
8.5.7	Content component attribute for Representation.....	192
8.6	Segment formats.....	192

8.6.1	Initialization Segment for OMAF base track.....	192
8.6.2	Tile Index Segment.....	196
8.6.3	Tile Data Segment.....	197
9	Omnidirectional media encapsulation and signalling in MMT	198
9.1	Architecture of MMT delivery in OMAF	198
9.2	OMAF signalling in MPEG composition information.....	199
9.3	VR application-specific MMT signalling.....	199
9.3.1	General	199
9.3.2	MMT signalling	200
10	Media profiles	216
10.1	Video profiles.....	216
10.2	Audio profiles	250
10.3	Image profiles.....	259
10.4	Timed text profiles	266
11	Presentation profiles.....	267
11.1	OMAF viewport-independent baseline presentation profile	267
11.1.1	General	267
11.1.2	ISO Base Media File Format constraints.....	267
11.2	OMAF viewport-dependent baseline presentation profile	268
11.2.1	General	268
11.2.2	ISO Base Media File Format constraints.....	268
12	OMAF toolset brands.....	268
12.1	Overlay toolset brand.....	268
12.1.1	Overview.....	268
12.1.2	ISO Base Media File Format constraints.....	268
12.1.3	OMAF player operation.....	269
12.2	Viewpoint toolset brand	269
12.2.1	Overview.....	269
12.2.2	ISO Base Media File Format constraints.....	269
12.2.3	OMAF player operation.....	269
12.3	Non-linear storyline toolset brand.....	269
12.3.1	Overview.....	269
12.3.2	ISO Base Media File Format constraints.....	269
12.3.3	OMAF player operation.....	270
	Annex A (normative) OMAF DASH schema	271
	Annex B (normative) DASH integration of media profiles	275
	Annex C (normative) CMAF integration of media profiles.....	288
	Annex D (informative) Viewport-dependent omnidirectional video processing.....	292
	Annex E (informative) DASH MPD examples	335
	Annex F (informative) MMT signalling examples.....	340
	Annex G (normative) Expected behaviour of OMAF player.....	342
	Bibliography	352

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This third edition cancels and replaces the second edition (ISO/IEC 23090-2:2021), which has been technically revised.

The main changes are as follows:

- The following profiles for the Versatile Video Coding (VVC) standard (Rec. ITU-T H.266 | ISO/IEC 23090-3) have been added:
 - VVC-based simple tiling OMAF video profile,
 - VVC-based viewport-independent OMAF video profile,
 - OMAF VVC image profile, and
 - CMAF media profile for the VVC-based viewport-independent OMAF video profile.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

When omnidirectional media content is consumed with a head-mounted display and headphones, only the parts of the media that correspond to the user's viewing orientation are rendered, as if the user were in the spot where and when the media was captured. One of the forms of omnidirectional media applications is omnidirectional video, also known as 360° video. Omnidirectional video is typically captured by multiple cameras that cover the entire sphere or at least a large part of the sphere. Compared to traditional media application formats, the end-to-end technology for omnidirectional video (from capture to playback) is more easily fragmented due to various capturing and video projection technologies. From the capture side, there exist many different types of cameras capable of capturing 360° video, and on the playback side there are many different devices that are able to playback 360° video with different processing capabilities. To avoid fragmentation of omnidirectional media content and devices, a standardized format for omnidirectional media applications is specified in this document.

This document defines a media format that enables omnidirectional media applications, focusing on 360° video, images, and audio, as well as associated timed text. What is specified in this document includes (but is not limited to):

- a coordinate system that consists of a unit sphere and three coordinate axes, namely the X (back-to-front) axis, the Y (lateral, side-to-side) axis, and the Z (vertical, up) axis;
- projection and rectangular region-wise packing methods that may be used for conversion of a spherical video sequence or image into a two-dimensional rectangular video sequence or image, respectively;
- storage of omnidirectional media and the associated metadata using the ISO Base Media File Format (ISOBMFF) as specified in ISO/IEC 14496-12;
- storage of video or image overlays and the associated metadata using ISOBMFF;
- encapsulation, signalling, and streaming of omnidirectional media and overlays in a media streaming system, e.g. dynamic adaptive streaming over HTTP (DASH) as specified in ISO/IEC 23009-1 or MPEG media transport (MMT) as specified in ISO/IEC 23008-1;
- media profiles and presentation profiles that provide conformance points for media codecs as well as media coding and encapsulation configurations that may be used for compression, streaming, and playback of the omnidirectional media content;
- toolset brands that provide conformance points for functionalities beyond plain 360° video, images and audio.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Coded representation of immersive media —

Part 2: Omnidirectional media format

1 Scope

This document specifies the omnidirectional media format for coding, storage, delivery and rendering of omnidirectional media, including video, images, audio and timed text. Omnidirectional image or video can contain graphics elements generated by computer graphics but encoded as image or video. Multiple viewpoints, each corresponding to an omnidirectional camera, are supported. The document also specifies storage and delivery of overlay images or video intended to be rendered over the omnidirectional background image or video.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10918-1, *Information technology — Digital compression and coding of continuous-tone still images — Part 1: Requirements and guidelines*

ISO/IEC 14496-1:2010, *Information technology — Coding of audio-visual objects — Part 1: Systems*

ISO/IEC 14496-3:2019, *Information technology — Coding of audio-visual objects — Part 3: Advanced audio coding*

Rec. ITU-T H.264 (08/21) | ISO/IEC 14496-10:2020, *Information technology — Coding of audio-visual objects — Part 10: Advanced video coding*

ISO/IEC 14496-12:2022, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 14496-14, *Information technology — Coding of audio-visual objects — Part 14, MP4 file format*

ISO/IEC 14496-15:2022, *Information technology — Coding of audio-visual objects — Part 15, Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format*

ISO/IEC 14496-30:2018, *Information technology — Coding of audio-visual objects — Part 30: Timed text and other visual overlays in ISO base media file format*

ISO/IEC 23000-19:—¹, *Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media*

ISO/IEC 23000-22:2019, *Information technology — Multimedia application format — Part 22 Multi-image application format (MIAF)*

Rec. ITU-T H.274 | ISO/IEC 23002-7, *Information technology — MPEG video technologies — Part 7: Versatile supplemental enhancement information messages for coded video bitstreams*

ISO/IEC 23003-3:2020, *Information technology — MPEG audio technologies — Part 3: Unified speech and audio coding*

ISO/IEC 23003-4:2020, *Information technology — MPEG audio technologies — Part 4: Dynamic range control*

ISO/IEC 23008-1:2017, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 1: MPEG media transport (MMT)*

Rec. ITU-T H.265 (11/19) | ISO/IEC 23008-2:2020, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

ISO/IEC 23008-3:2022, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 3: 3D audio*

ISO/IEC 23008-12, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image file format*

ISO/IEC 23009-1:2022, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

Rec. ITU-T H.266 (08/20) | ISO/IEC 23090-3:2021, *Information technology — Coded representation of immersive media — Part 3: Versatile video coding*

ISO/IEC 23091-2, *Information technology — Coding-independent code points — Part 2: Video*

ISO/IEC 23091-3, *Information technology — Coding-independent code points — Part 3: Audio*

IETF BCP 47, *Tags for Identifying Languages*

IETF Internet Standard 66, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 6381, *MIME Codecs and Profiles*

W3C Candidate Recommendation, *WebVTT: The Web Video Text Tracks Format*

W3C Recommendation, *TTML Profiles for Internet Media Subtitles and Captions 1.0.1 (IMSC1)*

W3C Recommendation, *XML schema part 1: Structures*

W3C Recommendation, *XML schema part 2: Datatypes*

¹ Under preparation. Stage at the time of publication: ISO/IEC DIS 23000-19:2022.

3 Terms, definitions, abbreviated terms and symbols

3.1 Terms and definitions

For the purposes of this document, the terms and definitions in ISO/IEC 14496-12, ISO/IEC 23008-12 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1 azimuth

first of the two sphere coordinates describing the location of a point on the sphere

Note 1 to entry: Azimuth and elevation are specified in subclause 5.1.

3.1.2 azimuth circle

circle on the sphere connecting all points with the same azimuth value

Note 1 to entry: An *azimuth circle* is always a *great circle* (3.1.22).

3.1.3 background visual media

piece of *visual media* (3.1.66) on which an *overlay* (3.1.34) is superimposed

3.1.4 circular image

image captured with a *fish-eye lens* (3.1.18)

3.1.5 closed scheme type

scheme type (3.1.49) that clearly specifies which transformations are allowed and does not allow future extensions

3.1.6 common reference coordinate system

3D Cartesian coordinate system with the centre being (X, Y, Z) equal to (0, 0, 0), used as the reference coordinate system for all *viewpoints* (3.1.63) within a *viewpoint group* (3.1.64)

3.1.7

composition-aligned sample

sample in a track that is associated with another track, the sample has the same composition time as a particular sample in the another track, or, when a sample with the same composition time is not available in the another track, the closest preceding composition time relative to that of a particular sample in the another track

3.1.8

composition picture

picture that is suitable to be presented and is obtained from the decoding outputs of *composition-aligned samples* (3.1.7) of all tracks of a 2D spatial relationship track group by arranging them spatially as specified by the semantics of the 2D spatial relationship track group

3.1.9

constituent picture

such part of a spatially frame-packed stereoscopic picture that corresponds to one view, or a picture itself when frame packing is not in use or the temporal interleaving frame packing arrangement is in use

3.1.10

content coverage

one or more *sphere regions* (3.1.51) that are covered by the content represented by the track, an image item, or a *composition picture* (3.1.8)

3.1.11

elevation

second of the two sphere coordinates describing the location of a point on the sphere

Note 1 to entry: Azimuth and elevation are specified in subclause 5.1.

3.1.12

elevation circle

circle on the sphere connecting all points with the same elevation value

Note 1 to entry: When the elevation is zero, an *elevation circle* is also a *great circle* (3.1.22). This coincides with the equator on Earth.

3.1.13

encoded tile sequence

coded representation of a *tile sequence* (3.1.56) that has the capability to be merged with other encoded tile sequences in coded domain without decoding mismatch by rewriting only header data

3.1.14

extractor track

track that has untransformed sample entry type equal to 'hvc2', 'avc2', or 'avc4' and contains one or more 'scal' track references

3.1.15

field of view

extent of the observable world in captured/recorded content or in a physical display device

3.1.16 file decoder

collective term for file/segment decapsulation and decoding of video, audio or image bitstreams

3.1.17 file decoding process

process specified as a part of a media profile specification that takes as input a set of ISOBMFF tracks or items and derives either a decoded pictures or audio samples, and rendering metadata for them; or a fully rendered audio scene in the reference system

3.1.18 fisheye lens

wide-angle camera lens that usually captures an approximately hemispherical *field of view* (3.1.15) and projects it as a *circular image* (3.1.4)

3.1.19 fisheye omnidirectional video fisheye omnidirectional image

omnidirectional media (3.1.32) where *circular images* (3.1.4) of *fisheye lenses* (3.1.18) are spatially arranged onto picture(s)

3.1.20 fisheye video

video captured by *fisheye lenses* (3.1.18)

3.1.21 global coordinate axes

coordinate axes that are associated with audio, video, and images representing the same acquisition position and intended to be rendered together

Note 1 to entry: Coordinate axes are specified in subclause 5.1.

Note 2 to entry: The origin of the global coordinate axes is usually the same as the centre point of a device or rig used for omnidirectional audio/video acquisition as well as the position of the observer's head in the three-dimensional space in which the audio and video tracks are located.

Note 3 to entry: In the absence of the initial viewing orientation metadata (see subclause 7.7.4 for tracks or subclause 7.9.9 for image items), the initial viewing orientation should be inferred to be equal to (0, 0, 0) for (centre_azimuth, centre_elevation, centre_tilt) relative to the global coordinate axes.

3.1.22 great circle

intersection of the sphere and a plane that passes through the centre point of the sphere

Note 1 to entry: A *great circle* is also known as an orthodrome or Riemannian circle.

Note 2 to entry: The centre of the sphere and the centre of a *great circle* are co-located.

3.1.23 guard band

area that is not rendered but may be used to improve the rendering quality to avoid or mitigate visual artifacts such as seams

Note 1 to entry: Guard bands in *packed pictures* (3.1.37) are associated with *packed regions* (3.1.38) as described in subclause 7.5.3.

3.1.24 header data

high-level syntax structures that do not contain arithmetic entropy-coded syntax elements

Note 1 to entry: In HEVC, parameter sets and slice segment headers can be regarded as header data. In VVC bitstreams with one or more independent VVC subpictures, parameter sets can be regarded as header data.

3.1.25 independent VVC subpicture

subpicture, as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3, that has `sps_subpic_treated_as_pic_flag[i]` equal to 1 and `sps_loop_filter_across_subpic_enabled_pic_flag[i]` equal to 0

Note 1 to entry: An independent VVC subpicture is treated like a picture in the VVC decoding process and loop filtering across the boundaries of an independent VVC subpicture is disabled.

3.1.26 local coordinate axes

coordinate axes obtained after applying rotation to the *global coordinate axes* (3.1.21)

3.1.27 mesh omnidirectional video

omnidirectional media (3.1.32) where rectangular regions of two-dimensional pictures are mapped to mesh elements of a three-dimensional mesh

3.1.28 OMAF base track

track whose samples are resolved by merging samples of the referenced *OMAF tile tracks* (3.1.30)

Note 1 to entry: An HEVC tile base track as specified in ISO/IEC 14496-15, an *extractor track* (3.1.14), and a VVC merge base track as specified in ISO/IEC 14496-15 that reference OMAF tile tracks are examples of OMAF base tracks.

3.1.29 OMAF player

application responsible for receiving files or segments or accessing files locally, decapsulating these files and segments, decoding the audio, video, image or timed text bitstreams, rendering the audio, video, image or timed text and applying a strategy for receiving the files or segments based on the *viewport* (3.1.65) information

3.1.30 OMAF tile bitstream

coded video data that represents a rectangular subset of the original video content and has the capability to be merged with other OMAF tile bitstreams in coded domain without decoding mismatch by rewriting only *header data* (3.1.24), where a decoding mismatch refers to the value of any pixel having a different value when decoded prior to merging or subsequently to merging

3.1.31 OMAF tile track

track that represents a rectangular subset of the original video content and has the capability to be merged with other OMAF tile tracks in coded domain without decoding mismatch by rewriting only header data, where a decoding mismatch refers to the value of any pixel having a different value when decoded prior to merging or subsequently to merging

Note 1 to entry: An HEVC tile track as specified in ISO/IEC 14496-15 and a VVC subpicture track as specified in ISO/IEC 14496-15 are examples of an OMAF tile track.

3.1.32 omnidirectional media

media such as image or video and its associated audio that enable rendering according to the user's *viewing orientation* (3.1.60), if consumed with a head-mounted device, or according to user's desired *viewport* (3.1.65), otherwise, as if the user was in the spot where and when the media was captured

3.1.33 open-ended scheme type

scheme type (3.1.49) that allows future extensions

3.1.34 overlay

piece of *visual media* (3.1.66) rendered over omnidirectional video or image item or over a *viewport* (3.1.65)

3.1.35 overlay source

piece of *visual media* (3.1.66) used as the content for an *overlay* (3.1.34)

3.1.36 overlay source region

overlay source (3.1.35) specified as a rectangular region within decoded pictures

3.1.37 packed picture

picture that is represented as a coded picture in the coded video bitstream

Note 1 to entry: A *packed picture* may result from *region-wise packing* (3.1.46) of a *projected picture* (3.1.40).

3.1.38 packed region

region in a *packed picture* (3.1.37) that is mapped to a *projected region* (3.1.41) as specified by the region-wise packing signalling

3.1.39 projected omnidirectional video projected omnidirectional image

omnidirectional media (3.1.32) where the relation of two-dimensional picture coordinates to *sphere coordinates* (3.1.50) is specified using mathematical projection equation(s)

3.1.40 projected picture

picture that has a representation format specified by an omnidirectional video projection format

Note 1 to entry: Omnidirectional projection formats are specified in subclause 5.2.

Note 2 to entry: A project picture i) covers the entire sphere. ii) has no re-sized, re-positioned, rotated, or mirrored regions relative to a representation format specified by an omnidirectional video projection format, and iii) has no *guard bands* (3.1.23).

3.1.41 projected region

region in a *projected picture* (3.1.40) that is mapped to a *packed region* (3.1.38) as specified by the region-wise packing signalling

3.1.42 projection

inverse of the process by which the samples of a *projected picture* (3.1.40) are mapped to a set of positions identified by a set of *azimuth* (3.1.1) and *elevation* (3.1.11) coordinates on a unit sphere

3.1.43 quality ranking region

region that is associated with a quality ranking value and is specified relative to a decoded picture or a sphere

3.1.44 quality ranking 2D region

quality ranking region (3.1.43) that is specified relative to a decoded picture

3.1.45 quality ranking sphere region

quality ranking region (3.1.43) that is specified relative to a sphere

3.1.46 region-wise packing

inverse of the process of transformation, resizing, and relocating of *packed regions* (3.1.38) of a *packed picture* (3.1.37) to remap to *projected regions* (3.1.41) of a *projected picture* (3.1.40)

3.1.47 rendering

process of generating audio-visual content for playback from the decoded audio-visual data according to the user's *viewing orientation* (3.1.60), if consumed with a head-mounted device, or according to user's desired *viewport* (3.1.65), otherwise, as well as optionally the user's *viewing position* (3.1.61) when sphere-relative *overlays* (3.1.34) are displayed with *background visual media* (3.1.3)

3.1.48 sample

all the data associated with a single time or single element in one of the three sample arrays that represent a picture

Note 1 to entry: When the term sample is used in the context of a track, it refers to all the data associated with a single time of that track, where a time is either a decoding time or a composition time. When the term sample is used in the context of a picture (e.g. in the phrase "luma sample") it refers to a single element in one of the three sample arrays that represent the picture.

3.1.49 scheme type

type that parameterizes an encrypted, restricted, or incomplete media track

3.1.50 sphere coordinates

azimuth (3.1.1) and *elevation* (3.1.11) that identify a location of a point on the unit sphere

3.1.51 sphere region

region on a sphere, specified either by four *great circles* (3.1.22) or by two *azimuth circles* (3.1.2) and two *elevation circles* (3.1.12), or such a region on the rotated sphere after applying certain amount of yaw, pitch, and roll rotations

3.1.52 sub-picture

picture that represents a spatial subset of the original video content, which has been split into spatial subsets before video encoding at the content production side

Note 1 to entry: One way to encode a sequence of independent VVC subpictures is to split the original video content into sub-pictures before encoding and encode a sub-picture sequence into a VVC bitstream having pictures containing only a single VVC subpicture with a specific subpicture identifier value.

3.1.53 sub-picture bitstream

bitstream that represents a spatial subset of the original video content, which has been split into spatial subsets before video encoding at the content production side

Note 1 to entry: It is possible to form a sub-picture bitstream from a sequence of independent VVC subpictures.

3.1.54 sub-picture track

track that is with spatial relationships to other track(s) and that represents a *sub-picture bitstream* (3.1.53)

Note 1 to entry: A VVC subpicture track, as specified in ISO/IEC 14496-15, is an example of a sub-picture track.

3.1.55 switch region

region that initiates viewpoint switching when selected by the user

Note 1 to entry: The following region types are specified for a switch region: a region on the viewport, a sphere region, and an overlay.

3.1.56 tile sequence

rectangular subset of the original video content

3.1.57 tilt angle

angle indicating the amount of tilt of a *sphere region* (3.1.51), measured as the amount of rotation of the *sphere region* along the axis originating from the sphere origin passing through the centre point of the *sphere region*, where the angle value increases clockwise when looking from the origin towards the positive end of the axis

3.1.58 time-parallel sample

sample in a track that is associated with another track, and either has the same decoding time as a particular sample in the other track, or, when a sample with the same decoding time in the other track is not available, the closest preceding decoding time relative to that of a particular sample in the other track

3.1.59 track sample entry type

sample entry type of a `SampleEntry` directly contained in `SampleDescriptionBox`

3.1.60 viewing orientation

triplet of *azimuth* (3.1.1), *elevation* (3.1.11), and *tilt angle* (3.1.57) characterizing the orientation that a user is consuming the audio-visual content; in case of image or video, characterizing the orientation of the *viewport* (3.1.65)

3.1.61 viewing position

position from which video and image content is viewed

Note 1 to entry: When there are no *overlays* (3.1.34), the viewing position is the centre of the unit sphere. A viewing position within the *viewing space* (3.1.62) is enabled to view *overlays* (3.1.34) from different perspectives relative to the *background visual media* (3.1.3). When the content is rendered with a head-mounted display, head tracking can be used to locate the viewing position.

3.1.62 viewing space

three-dimensional space of *viewing positions* (3.1.61) within which *rendering* (3.1.47) of image and video is enabled and VR experience is valid

3.1.63 viewpoint

omnidirectional media (3.1.32) corresponding to one omnidirectional camera

3.1.64 viewpoint group

group of *viewpoints* (3.1.63) that share the same *common reference coordinate system* (3.1.6)

3.1.65 viewport

region of *omnidirectional media* (3.1.32) suitable for display and viewing by the user

3.1.66**visual media**

video, image item or timed text

3.1.67**VVC subpicture**

subpicture as defined in Rec. ITU-T H.266 | ISO/IEC 23090-3

3.2 Abbreviated terms

2D	two-dimensional
3D	three-dimensional
CICP	coding-independent code points (specified in ISO/IEC 23091-1, 23091-2, and 23091-3)
CMAF	common media application format (specified in ISO/IEC 23000-19)
CMP	cubemap projection
DASH	MPEG dynamic adaptive streaming over HTTP (specified in ISO/IEC 23009-1)
ERP	equirectangular projection
FOV	field of view
GPS	global positioning system
ID	identifier
ISOBMFF	ISO base media file format (specified in ISO/IEC 14496-12)
HEVC	high efficiency video coding (specified in Rec. ITU-T H.265 ISO/IEC 23008-2)
HMD	head-mounted display
HOA	higher-order ambisonics
MCTS	motion-constrained tile set
MMT	MPEG media transport (specified in ISO/IEC 23008-1)
OMAF	omnidirectional media format (specified in ISO/IEC 23090-2)
SRD	spatial relationship description
URN	uniform resource name
VR	virtual reality
VSEI	versatile supplemental enhancement information (as specified in Rec. ITU-T H.274 ISO/IEC 23002-7)
VVC	versatile video coding (as specified in Rec. ITU-T H.266 ISO/IEC 23090-3)

WGS world geodetic system

3.3 Symbols

3.3.1 Arithmetic operators and mathematical functions

+	addition
−	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
x^y	exponentiation Specifies x to the power of y . In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.
/	integer division with truncation of the result toward zero For example, $7 / 4$ and $-7 / -4$ are truncated to 1 and $-7 / 4$ and $7 / -4$ are truncated to -1.
÷	division in mathematical equations where no truncation or rounding is intended
$\frac{x}{y}$	division in mathematical equations where no truncation or rounding is intended
$\sum_{i=x}^y f(i)$	summation of $f(i)$ with i taking all integer values from x up to and including y
$x \% y$	modulus Remainder of x divided by y , defined only for integers x and y with $x \geq 0$ and $y > 0$.

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases} \quad (1)$$

$\text{Asin}(x)$ trigonometric inverse sine function, operating on an argument x that is in the range of -1.0 to 1.0, inclusive, with an output value in the range of $-\pi/2$ to $\pi/2$, inclusive, in units of radians

$\text{Atan}(x)$ trigonometric inverse tangent function, operating on an argument x that is any real number, with an output value in the range of $-\pi/2$ to $\pi/2$, inclusive, in units of radians

$$\text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad \text{if } x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad \text{if } x < 0 \text{ \& \& } y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad \text{if } x < 0 \text{ \& \& } y < 0 \\ +\frac{\pi}{2} & ; \quad \text{if } x == 0 \text{ \& \& } y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{otherwise} \end{cases} \quad (2)$$

$\text{Cos}(x)$ trigonometric cosine function operating on an argument x in units of radians

$\text{Floor}(x)$ largest integer less than or equal to x

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5) \quad (3)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases} \quad (4)$$

$\text{Sin}(x)$ trigonometric sine function operating on an argument x in units of radians

$\text{Tan}(x)$ trigonometric tangent function operating on an argument x in units of radians

3.3.2 Order of operation precedence

The following notation is used to specify a range of values:

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

- Operations of a higher precedence are evaluated before any operation of a lower precedence.
- Operations of the same precedence are evaluated sequentially from left to right.

Table 1 specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

operations (with operands x, y, and z)
"x++", "x--"
"!x", "-x" (as a unary prefix operator)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (as a two-argument operator), " $\sum_{i=x}^y f(i)$ "
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

3.3.3 Range notation

The following notation is used to specify a range of values:

$x = y..z$ x takes on integer values starting from y to z , inclusive, with x , y , and z being integer numbers and z being greater than y .

3.3.4 Variables

This document derives variables that are named by a mixture of lower case and upper case letter and without any underscore characters. Variables starting with an upper case letter are derived for the current syntax structure and all depending syntax structures. Variables starting with an upper case letter may be used in the specification for dependent syntax structures without mentioning the originating syntax structure of the variable. Variables starting with a lower case letter are only used within the subclause in which they are derived.

3.3.5 Processes

Processes are used to describe the decoding of syntax elements. A process has a separate specification and invoking. All syntax elements and upper case variables that pertain to the current syntax structure and depending syntax structures are available in the process specification and invoking. A process specification may also have a

lower case variable explicitly specified as input. Each process specification has explicitly specified an output. The output is a variable that can either be an upper case variable or a lower case variable.

When invoking a process, the assignment of variables is specified as follows:

- If the variables at the invoking and the process specification do not have the same name, the variables are explicitly assigned to lower case input or output variables of the process specification.
- Otherwise (the variables at the invoking and the process specification have the same name), assignment is implied.

3.3.6 Syntax structures

Syntax structures in Clauses 6 and 7 are specified with and shall be interpreted according to the syntactic description language (SDL) specified in ISO/IEC 14496-1:2010, Clause 8, with the following change:

- Unlike specified in ISO/IEC 14496-1:2010, Clause 8, this document allows a variable declaration in *expression1* of a for loop `for(expression1; expression2; expression3)`. Such a variable declaration may be used for a loop index variable with a data type.

NOTE As specified in ISO/IEC 14496-1:2010, subclause 8.3.6, this document allows declaring a syntax element that is an individual element in an array. Such a declaration follows ISO/IEC 14496-1:2010, Rule A.2: `typespec name[[index]]`; which declares the *index*-th element of the array name as an individual syntax element having the data type *typespec*. In the context of this document `typespec name[[index]]` is used to refer to the index in the semantics and is otherwise equivalent to `typespec name`.

3.3.7 Conventions for indicating the number of boxes in tables

This subclause specifies conventions used in box tables (i.e. Table 31, Table 32, Table 33, Table 34, Table 35, Table 51, Table 52, Table 53, Table 54, Table 55, Table 56, Table B.1, Table B.2, and Table B.3) where the nesting and the number of boxes with given four-character codes are specified.

The Format Req. column in the tables specifies the number of boxes that shall be present as follows:

- “*” specifies that “zero or more” boxes may be present.
- “+” specifies that “one or more” boxes shall be present.
- “0/1” specifies that either zero or one boxes shall be present.
- “1” specifies that one and only one box shall be present.

In addition to boxes included in a box table, any other boxes may be present unless specifically disallowed.

4 Overview

4.1 Overall architecture

Figure 1 shows a typical content flow process for an omnidirectional media application.

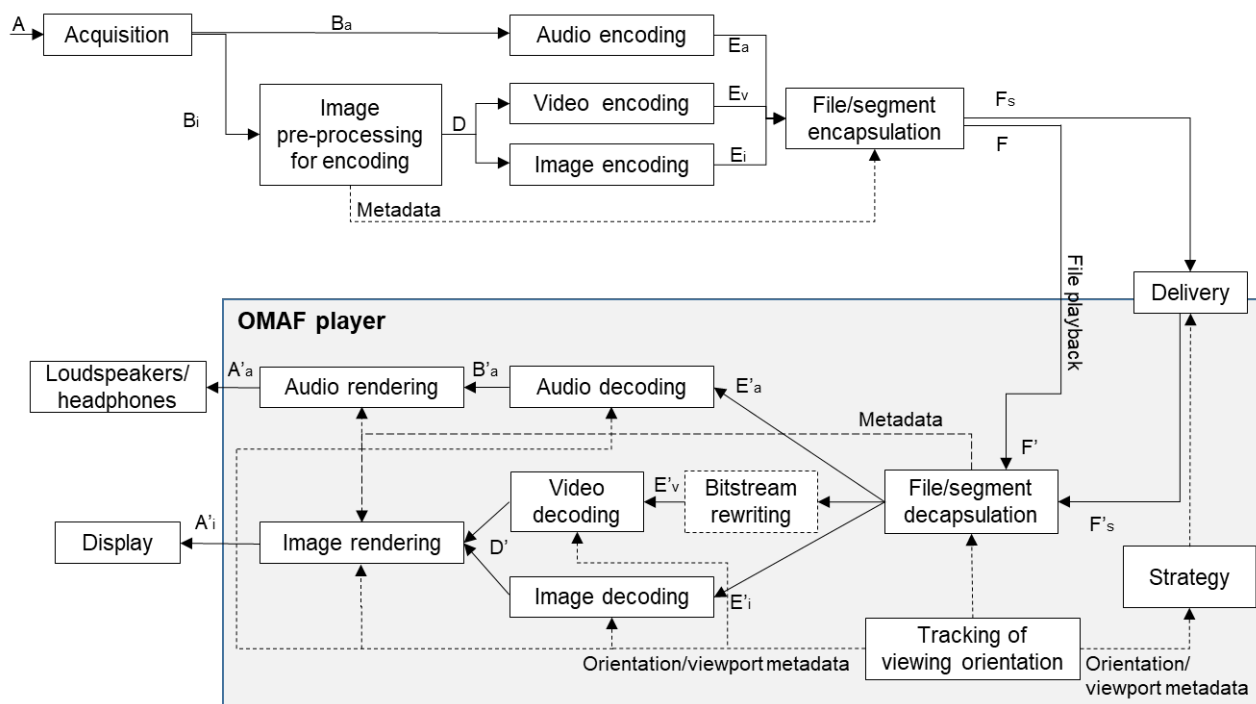


Figure 1 — Content flow process for omnidirectional media with projected video

The following interfaces are specified in this document:

- E'_a , E'_v , E'_i : audio bitstream, video bitstream, coded image(s), respectively; see Clause 10.
- F/F' : media file; see Clause 7. Moreover, media profiles specified in Clause 10 include the specification of the track formats for F/F' , which may contain constraints on the elementary streams contained within the samples of the tracks.
- Clause 8 specifies the delivery related interfaces for DASH delivery.
- Clause 9 specifies the delivery related interfaces for MMT delivery.

The other interfaces in Figure 1 are not specified in this document.

NOTE While the syntax and semantics of the bitstreams E_a , E_v , and E_i are the same as those for E'_a , E'_v , E'_i , respectively, the input interface to the file/segment encapsulation module is not specified.

A real-world audio-visual scene (A) is captured by audio sensors as well as a set of cameras or a camera device with multiple lenses and sensors. The acquisition results in a set of digital image/video (B_i) and audio (B_a) signals. The cameras/lenses typically cover all directions around the centre point of the camera set or camera device, thus the name of 360° video.

Audio may be captured using many different microphone configurations and stored as several different content formats, including channel-based signals, static or dynamic (i.e. moving through the 3D scene) object signals, and scene-based signals (e.g. Higher Order Ambisonics). The channel-based signals typically conform to one of the loudspeaker layouts defined in ISO/IEC 23091-3. In an omnidirectional media application, the loudspeaker layout signals of the rendered immersive audio program are binauralized for presentation via headphones.

For audio, no stitching process is needed, since the captured signals are inherently immersive and omnidirectional.

This document specifies the following types of omnidirectional video and images, which differ in the architecture in the image pre-preprocessing for encoding and in the image rendering processing blocks:

- Projected omnidirectional video/images:
 - Image pre-processing for encoding: The images (B_i) of the same time instance are stitched, possibly rotated, and projected onto a 2D picture coordinates using a mathematically specified projection format. Optionally, the resulting projected pictures may be further mapped region-wise onto a packed picture. Either projected pictures or packed pictures are subject to video or image encoding.
 - Image rendering: Either regions of the decoded packed pictures (if region-wise packing has been applied) or the entire projected picture (otherwise) is mapped onto a rendering mesh suitable for the projection format in use.
- Fisheye omnidirectional video/images:
 - Image pre-processing for encoding: Circular images (B_i) captured by fisheye lenses are arranged onto a 2D picture, which is then input to video or image encoding.
 - Image rendering: The decoded circular images are stitched using the signalled fisheye-specific parameters.
- Mesh omnidirectional video:
 - Image pre-processing for encoding: A 3D mesh consisting of mesh elements is generated, where mesh elements can be either parallelograms or regions of a sphere surface. The images (B_i) of the same time instance are stitched, possibly rotated, and projected onto the 3D mesh. Mesh elements are mapped onto rectangular regions of one or more 2D pictures, which are input to video encoding.
 - Image rendering: Rectangular regions of the decoded 2D picture(s) are mapped to the 3D mesh, which is used directly as the rendering mesh.

Further details of the architecture for projected, fisheye, and mesh omnidirectional video/images are provided in subclauses 4.2, 4.3, and 4.4, respectively.

The pre-processed pictures (D) are encoded as coded images (E_i) or a coded video bitstream (E_v). The captured audio (B_a) is encoded as an audio bitstream (E_a). The coded images, video or audio are then composed into a media file for file playback (F) or a sequence of an initialization segment and media segments for streaming (F_s), according to a particular media container file format. In this document, the media container file format is the ISO Base Media File Format specified in ISO/IEC 14496-12. The file encapsulator also includes metadata into the file or the segments.

The segments F_s are delivered using a delivery mechanism to a player.

The file that the file encapsulator outputs (F) is identical to the file that the file decapsulator inputs (F'). A file decapsulator processes the file (F') or the received segments (F'_s) and extracts the coded bitstreams (E'_a , E'_v , or E'_i) and parses the metadata. Viewport-dependent video may be carried in multiple tracks, which may be merged in the bitstream rewriting into a single video bitstream E'_v prior to decoding. The audio, video or images are then decoded into decoded signals (B'_a for audio, and D' for images/video). In the image rendering block, the decoded pictures (D') are projected onto the screen of a head-mounted display or any other display device based on the metadata parsed from the file. Likewise, decoded audio (B'_a) is rendered (e.g. through headphones) according to the current viewing orientation. The current viewing orientation is determined by the viewing orientation tracking functionality. When a head-mounted display is in use, the viewing orientation tracking can involve head tracking and possibly also eye tracking. When sphere-relative overlays are in use, the viewing orientation tracking functionality can include or be complemented by viewing position tracking and rendering of overlays with background visual media can take both the viewing position and the viewing orientation into account. Besides being used by the renderer to render the appropriate part of decoded video and audio signals, the current viewing orientation may also be used by the video and audio decoders for decoding optimization. In viewport-dependent delivery, the current viewing orientation is also passed to the strategy module, which determines the video tracks to be received based on the viewing orientation.

The process described above is applicable to both live and on-demand use cases.

4.2 Projected omnidirectional video/images

4.2.1 General

The architecture specified in subclause 4.1 applies with the following further specifications.

The images (B_i) of the same time instance are stitched, possibly rotated, projected, and mapped onto a packed picture (D). A description of this process is provided in subclause 4.2.2.

The packed pictures (D) are encoded as coded images (E_i) or a coded video bitstream (E_v).

The metadata in the file includes information for the following:

- the projection format of the projected picture;
- the area of the spherical surface covered by the packed picture;
- the rotation for converting between the global coordinate axes and the local coordinate axes;
- region-wise packing;
- region-wise quality ranking.

In the image rendering block, the decoded packed pictures (D') are projected onto the screen of a head-mounted display or any other display device based on the current viewing orientation or viewport and the projection, spherical coverage, rotation, and region-wise packing metadata parsed from the file.

4.2.2 Stitching, rotation, projection, and region-wise packing

For monoscopic 360° video, the source images of one time instance are stitched, possibly rotated, and projected to generate a projected picture representing one view. The breakdown of the image stitching, rotation, projection, and region-wise packing processes for monoscopic content is illustrated with Figure 2 and described as follows. Source images (B_i) are stitched, possibly rotated, and projected onto a unit sphere.

The image data on the unit sphere is further arranged onto a two-dimensional projected picture (C). The projected picture covers the entire sphere. Optionally, region-wise packing is then applied to map the projected picture onto a packed picture. If the region-wise packing is not applied, the packed picture is identical to the projected picture, and this picture is given as input to image/video encoding. Otherwise, projected regions are mapped onto a packed picture (D) by indicating the location, shape, and size of each packed region in the packed picture, and the packed picture (D) is given as input to image/video encoding. The packed picture may cover only a part of the entire sphere. In practice, the source images may be converted to a packed picture in one process without intermediate steps.

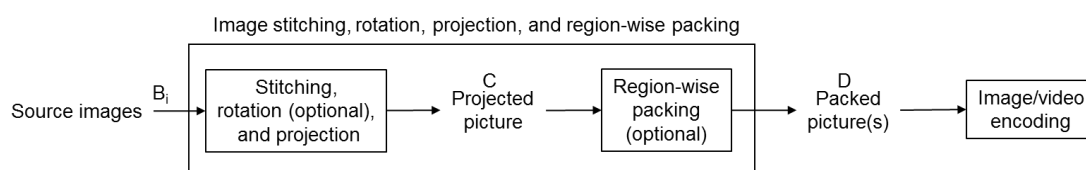


Figure 2 — Monoscopic image stitching, rotation, projection, and region-wise packing of source images of a single acquisition time instance

In the case of stereoscopic 360° video, the source images of one time instance are stitched, possibly rotated, and projected to generate a projected picture representing two views, one for each eye. Both views may be mapped onto the same packed picture, as described below in relation to Figure 3, and encoded by a traditional 2D video encoder. Alternatively, each view of the projected picture may be mapped to its own packed picture, in which case the image stitching, projection, and region-wise packing is like described above with Figure 2. A sequence of packed pictures of either the left view or the right view may be independently coded. Alternatively, packed pictures of the left view and the right view may be temporally interleaved in an alternating manner and encoded as a single bitstream, which is indicated to comply with the temporal interleaving frame packing arrangement. In another alternative, multiview coding is used to encode both views into the same bitstream with inter-view prediction.

The breakdown of the image stitching, rotation, projection, and region-wise packing processes for stereoscopic content where both views are mapped onto the same packed picture is illustrated with Figure 3 and described as follows. Source images (B_i) are stitched, possibly rotated, and projected onto two unit spheres, one for each eye. The image data on each unit sphere is further arranged onto a two-dimensional projected picture (C_L for left eye, C_R for right eye), which covers the entire sphere. Frame packing is applied to pack the left view picture and right view picture onto the same projected picture. Optionally, region-wise packing is then applied to pack the projected picture onto a packed picture, and the packed picture (D) is given as input to image/video encoding. If the region-wise packing is not applied, the packed picture is identical to the projected picture, and this picture is given as input to image/video encoding. The packed picture may cover only a part of the entire sphere.

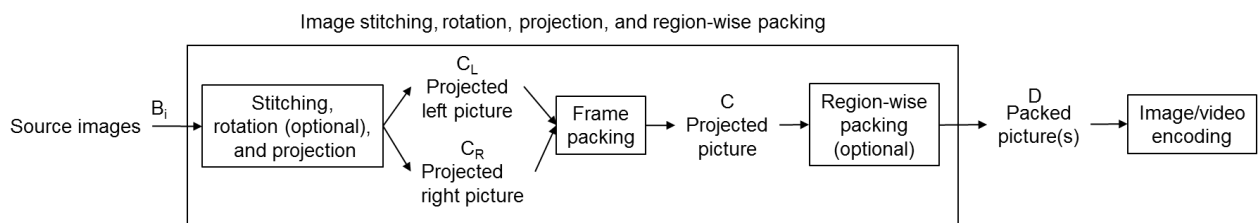


Figure 3 — Stereoscopic image stitching, rotation, projection, and region-wise packing of source images of a single acquisition time instance

The image stitching, rotation, projection, and region-wise packing processes may be carried out multiple times for the same source images to create different versions of the same content, e.g. for different sphere rotations (i.e. different rotations of local coordinate axes with respect to the global coordinate axes). Similarly, the region-wise packing process may be performed multiple times from the same projected picture to create more than one sequence of packed pictures to be encoded.

4.3 Fisheye omnidirectional video/images

The architecture specified in subclause 4.1 applies with the following further specifications.

A fisheye lens is a lens that achieves extremely wide angles of view and produces a circular image. Circular images captured by multiple fisheye lenses of a 360 camera typically cover all directions around the centre point of the camera set or camera device.

The circular images (B_i) captured by fisheye lenses at the same time instance are not stitched, but directly mapped onto a picture (D) in the fisheye video format, which is specified in Clause 6. Region-wise packing is not allowed for the fisheye video format.

The fisheye images including non-stitched circular images (D) are encoded as coded images (E_i) or a coded video bitstream (E_v). The coded images, video or audio are then composed into a media file for file playback (F) or a sequence of an initialization segment and media segments for streaming (F_s), according to a particular media container file format. The file encapsulator also includes fisheye-specific metadata, which assist in rendering the decoded pictures, into the file or the segments.

The fisheye-specific metadata in the file includes:

- Lens distortion correction (LDC) parameters with local variation of FOV,
- Lens shading compensation (LSC) parameters with RGB gains,
- Displayed field of view information, and
- Camera extrinsic parameters (physical location).

In an OMAF player, the circular images in the decoded fisheye video are stitched using the signalled fisheye-specific parameters to an omnidirectional image and rendered according to the user's viewport.

4.4 Mesh omnidirectional video

The architecture specified in subclause 4.1 applies with the following further specifications.

The images (B_i) of the same time instance are stitched, possibly rotated, projected onto a 3D mesh. A picture (D) to be encoded consists of rectangular regions, each mapping to exactly one mesh element of the 3D mesh.

The metadata in the file includes information for the following:

- the rotation for converting between the global coordinate axes and the local coordinate axes,
- the 3D mesh, consisting of mesh elements,
- a mapping of rectangular regions to mesh elements.

The decoded pictures (D') are mapped onto the 3D mesh used for rendering on a head-mounted display or any other display device based on the current viewing orientation or viewport.

4.5 Streaming methods for omnidirectional video

4.5.1 Overview

In viewport-independent streaming, omnidirectional video is represented with the equirectangular projection or another projection method. The video pictures are encoded as a single-layer bitstream using temporal inter prediction, the entire coded bitstream is stored at a server, and if needed, typically fully transmitted to the OMAF player, fully decoded by the decoder, and the area of the decoded picture corresponding to the current viewport is rendered to the user.

A distinct feature of omnidirectional video is that only a viewport is displayed at any particular time, while in normal video applications typically the entire video is displayed. This feature may be utilized to improve the performance of omnidirectional video systems, through selective delivery depending on the user's viewport (or any other criteria, such as recommended viewport timed metadata). Viewport-dependent delivery can be enabled for example by region-wise packing or viewport-dependent video coding. The performance improvement can be either or both of lower transmission bandwidth and lower decoding complexity compared to conventional omnidirectional video systems under the same resolution/quality of the video part presented to the user.

Annex D describes some specific schemes for viewport-dependent omnidirectional video processing, while subclause 4.5 includes an overview and categorization of viewport-dependent delivery.

Viewport-dependent delivery can be achieved by either of the following:

- Multiple versions of the content, where different versions are optimized for different viewing orientations. Examples of this category are provided in Clauses D.2 and D.9.
- Splitting the content into tile sequences. Each tile sequence can be encoded at multiple resolutions or at different quality levels. Viewport-dependent delivery can be achieved by receiving different sets of available encoded tile sequences.

The tile-based viewport-dependent streaming can be arranged in one of the following ways:

- a) Author-driven binding, which is further categorized into the following two methods:
 - 1) Viewport-specific author-driven binding: The content author prepares multiple OMAF base tracks, each targeted at a particular range of viewing orientations and providing bitstream rewriting instructions to merge specific encoded tile sequences. The strategy module selects between available OMAF base tracks as described in further detail in subclause 4.5.2. The bitstream rewriting module follows the instructions of the selected OMAF base track to create a video bitstream. Examples of this category are provided in subclauses D.6.2 to D.6.5, inclusive, and Clause D.8.
 - 2) Free-viewport author-driven binding. The content author prepares one or more OMAF base tracks. An OMAF base track is not tailor-made for any viewing orientation but gives multiple options from which the OMAF player can choose the encoded tile sequences that are used for extraction. Furthermore, an OMAF base track provides bitstream rewriting instructions to merge any encoded tile sequences selected by the OMAF player. The strategy module selects an OMAF base track and the encoded tile sequences among the indicated options. For example, an OMAF base track can enable the player to choose encoded tile sequences from multiple resolutions and the player can, for example, receive a full set of low-resolution tiles to show the full representation of the content at a lower quality while retrieving a subset of high-resolution tiles that together make up the viewport. The operation of the strategy module is described in subclause 4.5.3. The bitstream rewriting module follows the instructions of the selected OMAF base track to create a video bitstream. Examples of this category are provided in Clause D.4 and subclauses D.6.6 to D.6.10, inclusive.
- b) Late binding. The strategy module selects any encoded tile sequences that can be merged to a bitstream. It will typically retrieve a full set of low-resolution tiles to show the full representation of the content at a lower quality while retrieving a subset of high-resolution tiles that together make up the viewport. The tile selection strategy can be tailored according to the current viewport, available bandwidth, the field of view of the head-mounted display or the hardware decoding capabilities, among other things. The bitstream rewriting module merges the selected encoded tile sequences into a bitstream. The decoded picture/texture contains a combination of high and low resolution tiles which are then mapped onto a sphere, where the low quality tiles are used for those locations where no high-resolution tiles are being decoded. Further details on late binding are provided in subclause 4.5.4.

An ISOBMFF Media Segment as specified in ISO/IEC 23009-1 includes one or more `MovieFragmentBoxes` and one or more `MediaDataBoxes`. It is allowed to split an ISOBMFF Media Segment into Subsegments, each containing one or more `MovieFragmentBoxes` and one or more `MediaDataBoxes`. A DASH client either requests entire Media Segments through their URLs indicated in the MPD or entire Subsegments through their byte ranges indicated in the `SegmentIndexBox(es)`.

The duration of a Segment or a Subsegment can be too long for responding to a viewing orientation change with a new selection of OMAF tile tracks. To enable fine-grained requests, even down to a single picture interval, and to obtain the indexing data conveniently for all OMAF tile tracks, subclause 8.6 specifies the formats for the Initialization Segment for an OMAF base track, the Tile Index Segment, and the Tile Data Segment. The Initialization Segment for an OMAF base track contains not only a header for the OMAF base track but also a header for all the referenced OMAF tile tracks. The Tile Index Segment is an Index Segment as specified in ISO/IEC 23009-1 and contains `MovieFragmentBoxes` for the OMAF base track and all the referenced OMAF tile tracks. A Tile Data Segment contains media data for the respective OMAF tile track.

4.5.2 Tile-based streaming with viewport-specific author-driven binding

When an SRQR descriptor is present in one or more Main Adaptation Sets, an OMAF player concludes that the content has been made available for viewport-specific author-driven binding. An OMAF player chooses a Main Adaptation Set that provides a higher quality for the viewport as indicated by the SRQR descriptor. When there are many Main Adaptation Sets suitable for the viewport, the following factors can be used for selecting between them:

- Codec and the coding profile and level. These pieces of information are carried in the `@codecs` MPD attribute. The decoding capacity of the client device can constrain which Main Adaptation Sets are decodable in real-time.
- The restricted video scheme type carried in the `@codecs` MPD attribute or the projection format carried in the PF descriptor, or both.
- OMAF video profile. The OMAF video profile for the Main Adaptation Set is conveyed in the `@profiles` MPD attribute.
- Relative picture quality in the viewport area as indicated by the SRQR descriptor, when `sphRegionQuality@quality_ranking_local_flag` is equal to 0.
- The projected picture resolution from which the content on the viewport originates, as indicated by the SRQR descriptor. An OMAF player can select such Main Adaptation Set, which provides a suitably high resolution in relation to the resolution of the viewport.
- The region covered by the high-quality content, as indicated by the SRQR descriptor, in relation to the viewport size and margins for potential viewing orientation changes.

When a Partial Adaptation Set contains multiple Representations, the following factors can be used for selecting between them:

- Relative picture quality among the Representations as indicated by the SRQR descriptors. Alternatively, the `@bandwidth` value or the `@qualityRanking` value can be used as an indication of the respective quality among the Representations in the same Partial Adaptation Set.
- The bitrates for the Representations, as indicated with the `@bandwidth` MPD attribute, in relation to the estimated throughput.

4.5.3 Tile-based streaming with free-viewport author-driven binding

When an SRQR descriptor is absent in one or more Main Adaptation Sets, an OMAF player concludes that the content has been made available for free-viewport author-driven binding. When there are many Main Adaptation Sets, the following factors can be used for selecting between them:

- Codec and the coding profile and level. These pieces of information are carried in the `@codecs` MPD attribute. The decoding capacity of the client device can constrain which Main Adaptation Sets are decodable in real-time.
- The restricted video scheme type carried in the `@codecs` MPD attribute or the projection format carried in the PF descriptor, or both.
- OMAF video profile. The OMAF video profile for the Main Adaptation Set is conveyed in the `@profiles` MPD attribute.

An OMAF player can select the OMAF tile tracks to be received as follows:

- If the Segment formats specified in subclause 8.6 are in use, an OMAF player can use the following factors to select a subset of the OMAF tile tracks to be received:
 - The `TrackBoxes` of the OMAF tile tracks among which an OMAF player needs to select are present in the Initialization Segment of the Representation in the Main Adaptation Set.
 - The number of track references from the OMAF base track to OMAF tile tracks or to 'alte' track groups indicates the number of OMAF tile tracks that an OMAF player needs to select.
 - The `ContentCoverageBox` or the `SphereRegionQualityRankingBox` can be used to conclude which sphere region is covered by the OMAF tile track.
 - An OMAF player aims at selecting OMAF tile tracks covering the viewport so that they have a higher quality than the other selected OMAF tile tracks. The `@bandwidth` value or the `@qualityRanking` value can be used as an indication of the respective quality among the Representations in the same Partial Adaptation Set and hence to select between colocated OMAF tile tracks. Alternatively or additionally, the quality ranking metadata in the instances of `SphereRegionQualityRankingBox`, when present, can be used.
- Otherwise, an OMAF player selects a Representation from each Partial Adaptation Set. An OMAF player aims at selecting OMAF tile tracks covering the viewport so that they have a higher quality than the other selected OMAF tile tracks. When a Partial Adaptation Set contains multiple Representations of OMAF tile tracks, the following factors can be used for selecting between them:
 - Relative picture quality among the Representations as indicated by the SRQR descriptors. Alternatively, the `@bandwidth` value or the `@qualityRanking` value can be used as an indication of the respective quality among the Representations in the same Partial Adaptation Set.
 - The bitrates for the Representations, as indicated with the `@bandwidth` MPD attribute, in relation to the estimated throughput.

4.5.4 Tile-based streaming with late binding

4.5.4.1 Strategy

The client determines which tiles to retrieve. It will typically retrieve the full omnidirectional video in a low quality and the high-resolution tiles to provide high quality in the viewport. Although the strategy used by an OMAF player to determine which tiles to select at which point in time is up to the implementation, the following is a non-exhaustive list of elements that may be used in this process:

- Current viewport orientation (yaw, pitch, roll)
- Current zoom level (if any)
- The decoding capacity of the client-device. On a device with limited decoding capability, the tile selection algorithm may decide to limit the number of tiles it will fetch from the highest quality level. Or, alternatively, decide to skip a certain quality layer completely, and use a only fetch tiles from a medium quality level.
- Random access points for each tile
- Available network bandwidth
- Measured network latency
- The number of quality levels, and the number of tiles in each of those quality levels, along with their respective tile sizes

- Individual base tracks can be used to distinguish different quality levels.

A typical operation is to identify the tiles covering the current viewport (and the expected viewport in the next few hundred milliseconds). Based on the readings of the data generated by the user device, the client conceptually maps this target viewport onto the projected frame. The high-resolution tiles covering this area on the projected frame are then selected for retrieval. Upon reception, bitstream rewriting is performed on the retrieved tiles such that the rewritten bitstream contains the tiles for rendering the viewport, hence providing a partial view of the total projected frame. The client may, in addition, also decide to merge in the bitstream one or more low resolution tiles for covering the rest of the of projected frame.

4.5.4.2 Bitstream rewriting

4.5.4.2.1 General

A player can have many decoders at its disposal, a few, or only one. The most common case in devices at the time of writing this document is that a player has only one decoder, or just a few. To enable operation of devices that have only a single decoder, the bitstreams for the individual tiles need to be merged into a single bitstream.

Depending on device, viewport and network circumstances, it can be possible for a client deciding to select fewer tiles than are needed to satisfy the resolution the decoder has been initialized with, resulting in 'empty' tiles. In such a case, in order to still create a compliant bitstream, the client will have to fill those empty tiles with valid data. One method for doing so is to copy an already added tile as often as needed to make sure all tile 'slots' are filled. Another method is for a client to generate 'dummy' tile data.

ISO/IEC 14496-15 specifies that a VVC subpicture track that does not contain content intended for displaying (referred to as a substitute subpicture track) can be indicated with a box header flag of `VvcConfigurationBox` or `VvcNALUConfigBox` in the sample entry. A substitute subpicture track can be used to provide coded data for an 'empty' tile.

4.5.4.2.2 HEVC bitstream rewriting

The bitstream rewriting process is not specified in this document, with the present subclause giving directions on how to achieve such a rewrite for HEVC. Implementers are referred to Rec. ITU-T H.265 | ISO/IEC 23008-2 for more details on how to write a conformant bitstream.

Generating a conformant video bitstream typically consists of two types of rewriting that need to be carried out:

- a) Rewriting parameter sets in the Initialization Segment.
- b) Rewriting the slice segment header.

Additionally, it may be required to remove start code emulation prevention bytes before rewriting the slice segment header and insert the start code emulation prevention bytes after rewriting the slice segment header.

4.5.4.2.3 VVC bitstream rewriting

The bitstream rewriting process is not specified in this document, with the present subclause giving directions on how to achieve such a rewrite for VVC. Implementers can refer to Rec. ITU-T H.266 | ISO/IEC 23090-3 and ISO/IEC 14496-15 for more details on how to write a conformant bitstream.

Generating a conformant video bitstream typically consists of rewriting the parameter sets in the Initialization Segment.

The bitstream rewriting process could be similar to ISO/IEC 14496-15:2022, subclause 11.6, with the following differences:

- Rather than using the sequence parameter sets of the VVC merge base track, the player generates the sequence parameter sets that are suitable for the bitstream.

- The player picks OMAF tile tracks according to its strategy rather than using the 'spor' sample group in the VVC merge base track as specified in ISO/IEC 14496-15:2022, subclause 11.6.4.
- The player updates parameter sets based on the high-level syntax of the NAL units in the selected OMAF tile tracks rather than using the VVC merge base track and its sample groups for parameter set rewriting as specified in ISO/IEC 14496-15:2022, subclause 11.6.5.

It may be required to remove start code emulation prevention bytes before rewriting a parameter set and insert the start code emulation prevention bytes after rewriting the parameter set.

4.6 Additional functionalities

In addition to omnidirectional video, images and audio, this document specifies the following additional functionalities:

- Multiple viewpoints. A viewpoint can be regarded as an omnidirectional camera. Viewpoints can be used, for example, for the following purposes:
 - Provide several user-switchable camera positions to view the content. For example, a sports event can be captured from different positions.
 - Express a storyline where the user is given the choice to select which storyline path is followed. Each temporally contiguous portion in the storyline is represented by a viewpoint.

The storage of viewpoints is specified in subclause 7.12.

- Overlays. Video, image, or timed text can be indicated to be overlays of the omnidirectional background visual video or image. Overlays can be indicated to have sphere-relative or viewport-relative positions. The storage of overlays is specified in subclause 7.14.

The use of the additional functionalities can be indicated using the toolset brands specified in Clause 12.

4.7 Conformance and interoperability

4.7.1 General

This document specifies several referenceable features that implementations and external specifications may use to achieve interoperability. These referenceable features include the following:

- Specification of a coordinate system for omnidirectional media (subclause 5.1), and equations for a conversion process between sets of coordinate axes of this coordinate system (subclause 5.3).
- Omnidirectional projection formats (subclause 5.2).
- Media profiles, as defined in subclause 4.7.2. Individual media profiles are specified in Clause 10.
- Presentation profiles, as defined in subclause 4.7.3. Individual presentation profiles are specified in Clause 11.
- Mapping of media and presentation profiles to DASH and CMAF are specified in Annex B and Annex C, respectively. When conformance to the media or presentation profiles specified in this document is indicated for content carried according to ISO/IEC 23009-1, the specifications of Annex B shall be obeyed. When conformance to the media or presentation profiles specified in this document is indicated for content carried according to ISO/IEC 23000-19, the specifications of Annex C shall be obeyed.

- Toolset brands, as defined in subclause 4.7.4. Individual toolset brands are specified in Clause 12.
- Region-wise packing formats (subclause 5.4).
- Scheme types for post-decoder processing of omnidirectional video (subclause 7.6.1).
- Specific types of sphere region timed metadata tracks (subclause 7.7).

However, the technologies and features specified in this document may be used independently. For example, boxes specified in this document may be used independently of other OMAF features.

NOTE For implementing a media profile, it is suggested to refer to the specification of that media profile in Clause 10 and read other parts of this document selectively by following the cross-references included in the specification of the media profile.

This document specifies code points that implementations and external specifications may use for referring to respective OMAF features. The code points are summarized in subclause 4.7.5.

4.7.2 Media profiles

4.7.2.1 General

A media profile for timed media is defined as requirements and constraints for a set of one or more ISOBMFF tracks of a single media type. The conformance of a set of one or more ISOBMFF tracks to a media profile is specified as a combination of:

- Specification of which sample entry type(s) are allowed, and which constraints and extensions are required in addition to those imposed by the sample entry type(s).
- Constraints on the samples of the tracks, typically expressed as constraints on the elementary stream contained within the samples of the tracks.

A media profile for static media is defined as requirements and constraints for a set of one or more ISOBMFF items of a single media type. The conformance of a set of one or more ISOBMFF items to a media profile is specified as a combination of:

- Specification of which item type(s) are allowed, and which constraints and extensions are required in addition to those imposed by the item type(s).
- Constraints on the content of the items, typically expressed as constraints on the elementary stream contained within the items.

The elementary stream constraints of a media profile may be indicated by a requirement to comply with a certain profile and level of the media coding specification, possibly including additional constraints and extensions, such as a requirement of the presence of certain information for rendering and presentation.

NOTE The use of the elementary stream constraints specified for a media profile can be referenced by implementations and external specifications independently of the ISOBMFF requirements and constraints specified for the media profile. However, the encapsulation and use of a media profile for other encapsulation formats is outside of the scope of this document.

4.7.2.2 File decoding process and file decoder requirements for video and image media profiles

Each video or image media profile specified in Clause 10 includes a file decoding process such that all file decoders that conform to the video or image media profile will produce numerically identical cropped decoded pictures when invoking the file decoding process associated with that video or image media profile for a set of ISOBMFF tracks conforming to the video media profile or a set of ISOBMFF image items conforming to the image media profile, respectively. A bitstream that conforms to the elementary stream constraints specified for the video or

image media profile is reconstructed as an intermediate product of the file decoding process. The output of the file decoding process consists of all of the following:

- a list of decoded pictures;
- for projected omnidirectional video or image, all of the following rendering metadata:
 - the projection format of the associated projected pictures,
 - the frame packing format of the associated projected pictures (when applicable),
 - the region-wise packing information (when applicable),
 - the rotation information (when applicable);
- for fisheye omnidirectional video or image, fisheye-specific rendering metadata;
- for mesh omnidirectional video or image, all of the following metadata:
 - a 3D mesh, consisting of mesh elements,
 - a mapping of rectangular regions in the decoded pictures to mesh elements,
 - the rotation information (when applicable).

Video media profiles may specify constraints on when rendering metadata is allowed to change.

A file decoder conforms to the file decoding process requirements of this document when it complies with both of the following:

- The file decoder includes a conforming decoder that produces numerically identical cropped decoded pictures to those produced by the file decoding process specified for the video or image media profile in Clause 10 (with the correct output order or output timing, as specified in the video or image coding specification of the video or image media profile, respectively).
- The file decoder outputs rendering metadata that is equivalent to that produced by the file decoding process specified for the video or image media profile in Clause 10 (with the correct association of the rendering metadata to particular cropped decoded pictures, as specified in this document).

Figure 4 illustrates an example file decoder, its outputs as described above, and its relation to other parts of an OMAF player implementation.

A player claiming conformance to a video or image media profile shall include a file decoder complying with the file decoding process of that video or image media profile as specified above. Specifications of a media profile may include a subclause on expectations of a player operation, for example including recommendations for rendering.

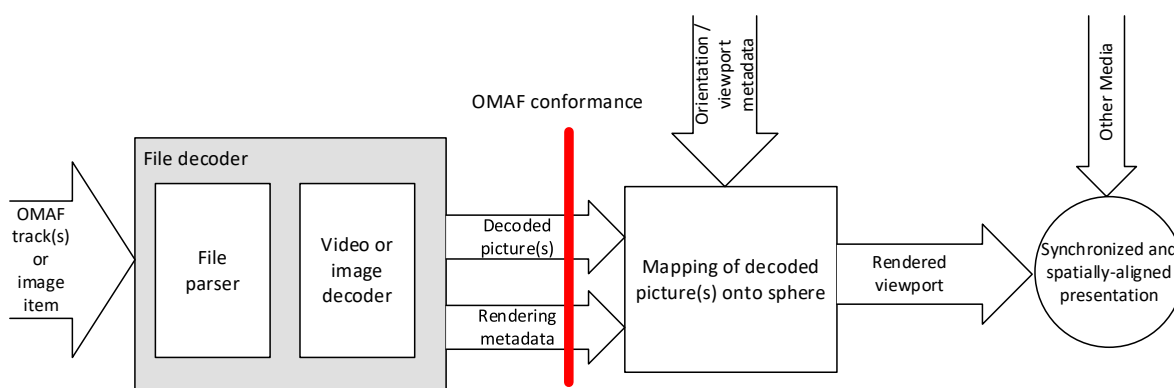


Figure 4 — Conformance points for OMAF video and image media profiles

4.7.2.3 File decoding process and file decoder requirements for audio media profiles

Each audio profile specified in Clause 10 includes a file decoding process such that all file decoders that conform to the audio profile will produce an output according to the specification of the file decoding process. For audio, the conformance of the output signal is defined by the file decoding process. The output of the file decoding process provides a signal that may be represented in the reference system.

4.7.3 Presentation profiles

A presentation profile is defined as requirements and constraints for an ISOBMFF file containing tracks or items of any number of media types. A specification of a presentation profile should refer to the specified media profiles and may include additional requirements or constraints. A file conforming to a presentation profile typically provides an omnidirectional audio-visual experience.

4.7.4 Toolset brands

A toolset brand facilitates a functionality that is considered additional compared to the basic operation of an OMAF player. A toolset brand should be specified so that the toolset brand can be used with any media coding format or media profile. A toolset brand is specified as requirements and constraints for a set of one or more ISOBMFF tracks and items. A file is indicated to conform to a toolset brand with a `FileTypeInfoBox` containing the four-character code of the toolset brand among `compatible_brands`. The specification of a toolset brand may include requirements for an OMAF player complying with the toolset brand.

4.7.5 Summary of referenceable code points

4.7.5.1 Brands

ISO/IEC 14496-12 defines the concept of brands, which may be indicated in a file-specific manner in the `FileTypeInfoBox` and in a track-specific manner in the `TrackTypeInfoBox`. Brands are used in this document to indicate conformance to OMAF media profiles, OMAF presentation profiles, OMAF toolset brands, and CMAF media profiles that are defined in this document.

The brands specified in this document are listed in Table 2.

Table 2 — Brands specified in this document

Brand identifier	Subclause in this document	Name of the media profile or the presentation profile
hevi	10.1.2	HEVC-based viewport-independent OMAF video profile
hevd	10.1.3	HEVC-based viewport-dependent OMAF video profile
avde	10.1.4	AVC-based viewport-dependent OMAF video profile
uhvi	10.1.5	Unconstrained HEVC-based viewport-independent OMAF video profile
adti	10.1.6	Advanced tiling OMAF video profile
siti	10.1.7	HEVC-based simple tiling OMAF video profile
sitv	10.1.7	VVC-based simple tiling OMAF video profile
vvci	10.1.8	VVC-based viewport-independent OMAF video profile
oabl	10.2.2	OMAF 3D audio baseline profile
oa2d	10.2.3	OMAF 2D audio legacy profile
heoi	10.3.3	OMAF HEVC image profile
jpoi	10.3.4	OMAF legacy image profile
vvoi	10.3.5	OMAF VVC image profile
ttml	10.4.2	OMAF IMSC1 timed text profile
ttwv	10.4.3	OMAF WebVTT timed text profile
ompp	11.1	OMAF viewport-independent baseline presentation profile
ovdp	11.2	OMAF viewport-dependent baseline presentation profile
ovly	12.1	Overlay toolset brand
vwpt	12.2	Viewpoint toolset brand
nls1	12.3	Non-linear storyline toolset brand
cvid	C.1.1	CMAF media profile for the HEVC-based viewport-independent OMAF video profile
chev	C.1.2	CMAF media profile for the HEVC-based viewport-dependent OMAF video profile
cuvd	C.1.3	CMAF media profile for the unconstrained HEVC-based viewport-independent OMAF video profile

Brand identifier	Subclause in this document	Name of the media profile or the presentation profile
cvvc	C.1.4	CMAF media profile for the VVC-based viewport-independent OMAF video profile
cabl	C.2.1	CMAF media profile for OMAF 3D audio baseline profile

4.7.5.2 Uniform resource names

The URNs specified in this document are listed in Table 3.

Table 3 — URNs specified in this document

URN	Subclause in this document	Description
urn:mpeg:mpegI:omaf:2017	8.2.2	Namespace for XML elements and attributes
urn:mpeg:mpegI:omaf:2017:pf	8.3.2	Scheme identifier for the omnidirectional projection type DASH MPD descriptor
urn:mpeg:mpegI:omaf:2017:rwpk	8.3.3	Scheme identifier for the region-wise packing type DASH MPD descriptor
urn:mpeg:mpegI:omaf:2017:cc	8.3.4	Scheme identifier for the content coverage DASH MPD descriptor
urn:mpeg:mpegI:omaf:2017:srqr	8.3.5	Scheme identifier for the spherical region-wise quality ranking DASH MPD descriptor
urn:mpeg:mpegI:omaf:2017:2dqr	8.3.6	Scheme identifier for the 2D region-wise quality ranking DASH MPD descriptor
urn:mpeg:mpegI:omaf:2017:fomv	8.3.7	Scheme identifier for the fisheye omnidirectional video DASH MPD descriptor
urn:mpeg:mpegI:omaf:2020	8.5.1	Namespace for XML elements and attributes
urn:mpeg:mpegI:omaf:2020:assoc	8.5.2	Scheme identifier for the association descriptor
urn:mpeg:mpegI:omaf:2020:vwpt	8.5.3	Scheme identifier for the viewpoint information descriptor
urn:mpeg:mpegI:omaf:2020:ovly	8.5.5	Scheme identifier for the overlay information descriptor
urn:mpeg:mpegI:omaf:2020:etgb	8.5.6	Scheme identifier for the entity to group descriptor

URN	Subclause in this document	Description
urn:mpeg:mpegI:omaf:dash:profile:indexed-isobmff:2020	B.1.4.1	DASH profile identifier indicating the use of the segment formats specified in subclause 8.5

4.7.5.3 Restricted scheme types

The restricted scheme types specified in this document are listed in Table 4.

Table 4 — Restricted scheme types specified in this document

Restricted scheme type	Subclause in this document	Description
podv	7.6.1.2	Projected omnidirectional video; an open-ended restricted scheme type for omnidirectional video based on a projection
erpv	7.6.1.3	Equirectangular projected video; a closed restricted scheme type for omnidirectional video based on the equirectangular projection and essentially no region-wise packing.
ercm	7.6.1.4	Packed equirectangular or cubemap projected video; a closed restricted scheme type for omnidirectional video based on either the equirectangular projection or the cubemap projection and any region-wise packing.
fodv	7.6.1.5	Fisheye omnidirectional video; an open-ended restricted scheme type for omnidirectional video based on video captured by one or more fisheye camera lenses and not based on a projection
ecov	7.6.1.6	Like 'ercm' but overlay information may additionally be present.
erc2	7.6.1.7	Like 'ercm' but the region-wise packing format of the track containing the 'erc2' scheme type is derived from the collection of the instances of <code>RegionWisePackingBox</code> of the tracks referenced by the track containing the 'erc2' scheme type.
modv	7.6.1.8	Mesh omnidirectional video; an open-ended restricted scheme type for omnidirectional video based on a 3D mesh
meov	7.6.1.9	Closed scheme type for mesh omnidirectional video

4.7.5.4 Sample entry types

The sample entry types specified in this document are listed in Table 5.

Table 5 — Sample entry types specified in this document

Sample entry type	Subclause in this document	Description
invo	7.7.4	For use with a timed metadata track containing the initial viewing orientation timed metadata
rcvp	7.7.5	For use with a timed metadata track containing the recommended viewport timed metadata without indicating a viewpoint
rvp2	7.7.5	For use with a timed metadata track containing the recommended viewport timed metadata concerning one or more indicated viewpoints
ttsl	7.7.7	For use with a timed metadata track containing the timed text sphere location timed metadata
stmd	7.11	For use with a timed metadata track containing ERP region metadata
dyvp	7.12.3.1	For use with a timed metadata track indicating the viewpoint parameters that are dynamically changing over time
invp	7.12.3.2	For use with a timed metadata track indicating the initial viewpoint that should be used
ocpc	7.12.3.3	For use with a timed metadata track indicating information on object centre points correspondence between viewpoints
dyol	7.14.6	For use with a timed metadata track indicating overlays that are active only at particular times and dynamically changing overlay parameters
vrsp	7.15.5	For use with a timed metadata track indicating viewing space

4.7.5.5 Box types

The box types specified in this document are listed in Table 6.

Table 6 — Box types specified in this document and their relation to boxes not specified in this document

Box types, structure and cross-reference.											
moov										ISOBMFF	container for all the metadata
	trak									ISOBMFF	container for an individual track or stream
		trgr								ISOBMFF	track grouping indication
			2dsr							7.1.3	spatial relationship 2D description box
				2dss						7.1.3	spatial relationship 2D source box
				sprg						7.1.3	sub-picture region box

Box types, structure and cross-reference.											
			coif							7.13.3	composition information box
		ttyp								ISOBMFF	track type box
		mdia								ISOBMFF	container for the media information in a track
			minf							ISOBMFF	media information container
				stbl						ISOBMFF	sample table box, container for the time/space map
					stsd					ISOBMFF	sample descriptions (codec types, initialization etc.)
					-					ISOBMFF	sample entry or restricted sample entry
						rinf				ISOBMFF	restricted scheme info box
							schi			ISOBMFF	scheme information box
								povd		7.6.2	projected omnidirectional video box
									prfr	7.6.2	projection format box
									rwpk	7.6.4	region-wise packing box
									rotn	7.6.5	rotation box
									covi	7.6.6	coverage information box
									ovly	7.14.4	overlay configuration box
								fovvd		7.6.3	fish-eye omnidirectional video box
								fovi		7.6.3	fish-eye video essential info box
								fvsi		7.6.3	fish-eye video supplemental info box
								movd		7.6.7	mesh omnidirectional video box
									mesh	7.6.8	mesh box
									rotn	7.6.5	rotation box
						-				7.7.2	sphere region sample entry
						rosc				7.7.2	sphere region configuration box
						rcvp				7.7.5	recommended viewport sample entry
							rvif			7.7.5	recommended viewport information box
						-				ISOBMFF	visual sample entry
							srqr			7.8.2	sphere region quality ranking box
							2dqr			7.8.3	2D region quality ranking box
							ovly			7.14.4	overlay configuration box
							vssn			7.15.3	viewing space box
						stpp				ISOBMFF	XML subtitle sample entry
							otcf			7.10.2	OMAF timed text configuration box
						wvtt				ISO/IEC 14496-30	WebVTT sample entry
							otcf			7.10.2	OMAF timed text configuration box
meta										ISOBMFF	Metadata
	iprp									ISOBMFF	item properties box

Box types, structure and cross-reference.											
		ipco								ISOBMFF	<i>item property container box</i>
			prfr							7.9.3	<i>projection format item property</i>
			fovi							7.9.4	<i>essential fisheye image item property</i>
			fvsi							7.9.5	<i>supplemental fisheye image item property</i>
			rwpk							7.9.6	<i>region-wise packing item property</i>
			rotn							7.9.7	<i>rotation item property</i>
			covi							7.9.8	<i>coverage information item property</i>
			stvi							7.9.2	<i>frame packing item property</i> NOTE FramePackingProperty has the same box type and syntax as StereoVideoBox specified in ISO/IEC 14496-12.
			liivo							7.9.9	<i>initial viewing orientation item property</i>
			ovly							7.14.5	<i>overlay item property</i>
			vssn							7.15.4	<i>viewing space item property</i>
meof										7.1.4	<i>media offset box</i>

4.7.5.6 Track grouping types

The track grouping types specified in this document are listed in Table 7.

Table 7 — Track grouping types specified in this document

Track grouping type	Subclause in this document	Track grouping name
2dsr	7.1.3.2	Two dimensional spatial relationship track grouping

4.7.5.7 Entity grouping types

The entity grouping types specified in this document are listed in Table 8.

Table 8 — Entity grouping types specified in this document

Entity grouping type	Subclause in this document	Entity grouping name
vipo	7.12.2	Viewpoint entity grouping
oval	7.14.7.1	Grouping of overlays that are alternatives for switching
ovbg	7.14.7.2	Grouping of overlays and background visual media that are intended to be presented together

4.7.5.8 Sample grouping types

The sample grouping types specified in this document are listed in Table 9.

Table 9 — Sample grouping types specified in this document

Sample grouping type	Subclause in this document	Sample grouping name
2dsr	7.1.3.3	Two dimensional spatial relationship sample grouping
tmsb	7.16.2	Tile mesh sample grouping

5 Omnidirectional video projection and region-wise packing

5.1 Coordinate system

The coordinate system consists of a unit sphere and three coordinate axes, namely the X (back-to-front) axis, the Y (lateral, side-to-side) axis, and the Z (vertical, up) axis, where the three axes cross at the centre of the sphere.

The location of a point on the sphere is identified by a pair of sphere coordinates azimuth (ϕ) and elevation (θ).

Figure 5 specifies the relation of the sphere coordinates azimuth (ϕ) and elevation (θ) to the X, Y, and Z coordinate axes.

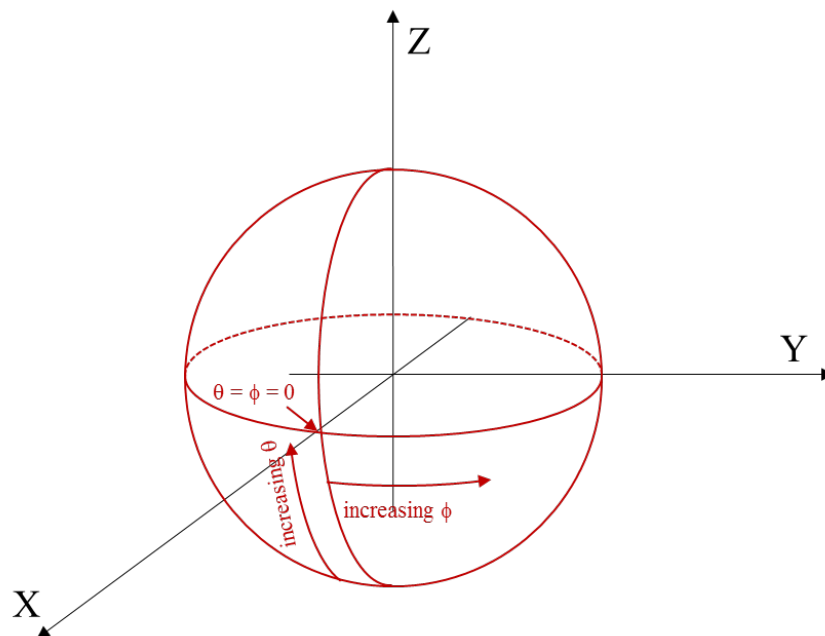


Figure 5 — Coordinate axes and their relation to the sphere coordinates

NOTE The specified coordinate system is the same as in ISO/IEC 23008-3.

The value range of azimuth is -180.0° , inclusive, to 180.0° , exclusive. The value range of elevation is -90.0° to 90.0° , inclusive.

5.2 Omnidirectional projection formats

5.2.1 General

Subclause 5.2 specifies the inverse of the projection process for remapping of one sample location of a monoscopic projected luma picture onto a position on the unit sphere identified by a pair of azimuth and elevation coordinates.

The omnidirectional projection formats specified in this document are identified by a 5-bit unsigned integer value. Table 10 specifies the omnidirectional projection format identifier values and provides references to the subclauses in which the respective inverse projection processes are specified.

Table 10 — Omnidirectional projection formats

Identifier value	Omnidirectional projection	Subclause
0	Equirectangular projection	5.2.2
1	Cubemap projection	5.2.3
2..31	Reserved	N/A

5.2.2 Equirectangular projection for one sample location

Inputs to this process are:

- pictureWidth and pictureHeight, which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units, and
- the centre point of a sample location (hPos, vPos) along the horizontal and vertical axes, respectively, where hPos and vPos are in relative projected picture sample units and may have non-integer real values.

Outputs of this process are:

- sphere coordinates (ϕ , θ) for the sample location in degrees relative to the coordinate axes specified in subclause 5.1.

The sphere coordinates (ϕ , θ) for the luma sample location, in degrees, are given by the following equations:

$$\begin{aligned}\phi &= (0.5 - \text{hPos} \div \text{pictureWidth}) * 360 \\ \theta &= (0.5 - \text{vPos} \div \text{pictureHeight}) * 180\end{aligned}\tag{5}$$

Figure 6 illustrates the azimuth and elevation ranges of a monoscopic projected picture with the equirectangular projection.

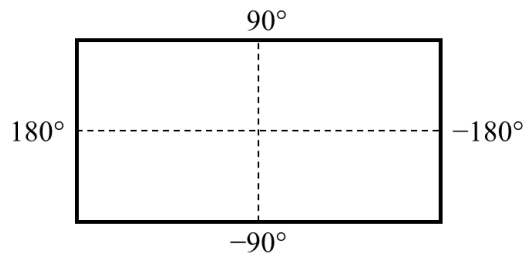


Figure 6 — Azimuth and elevation ranges of the monoscopic projected picture of the equirectangular projection

NOTE 1 Since an input to this process is the centre point of a sample location and the width and height of the sample are non-zero, the output ϕ is never equal to -180° or 180° and the output θ is never equal to -90° or 90° .

NOTE 2 The monoscopic projected picture represents the inside surface of the unit sphere observed from the origin of the coordinate system. Thus, the azimuth decreases from left to right.

5.2.3 Cubemap projection for one sample location

Inputs to this process are:

- `pictureWidth` and `pictureHeight`, which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units, and
- the centre point of a sample location (`hPos`, `vPos`) along the horizontal and vertical axes, respectively, where `hPos` and `vPos` are in relative projected picture sample units and may have non-integer real values.

Outputs of this process are:

- sphere coordinates (ϕ, θ) for the sample location in degrees relative to the coordinate axes specified in subclause 5.1.

Figure 7 illustrates the cube face arrangement in the projected picture of the cubemap projection format and the mapping of the cube faces onto the coordinate axes specified in subclause 5.1, where `PX`, `NX`, `PY`, `NY`, `PZ`, and `NZ` denote positive X, negative X, positive Y, negative Y, positive Z, and negative Z, respectively.

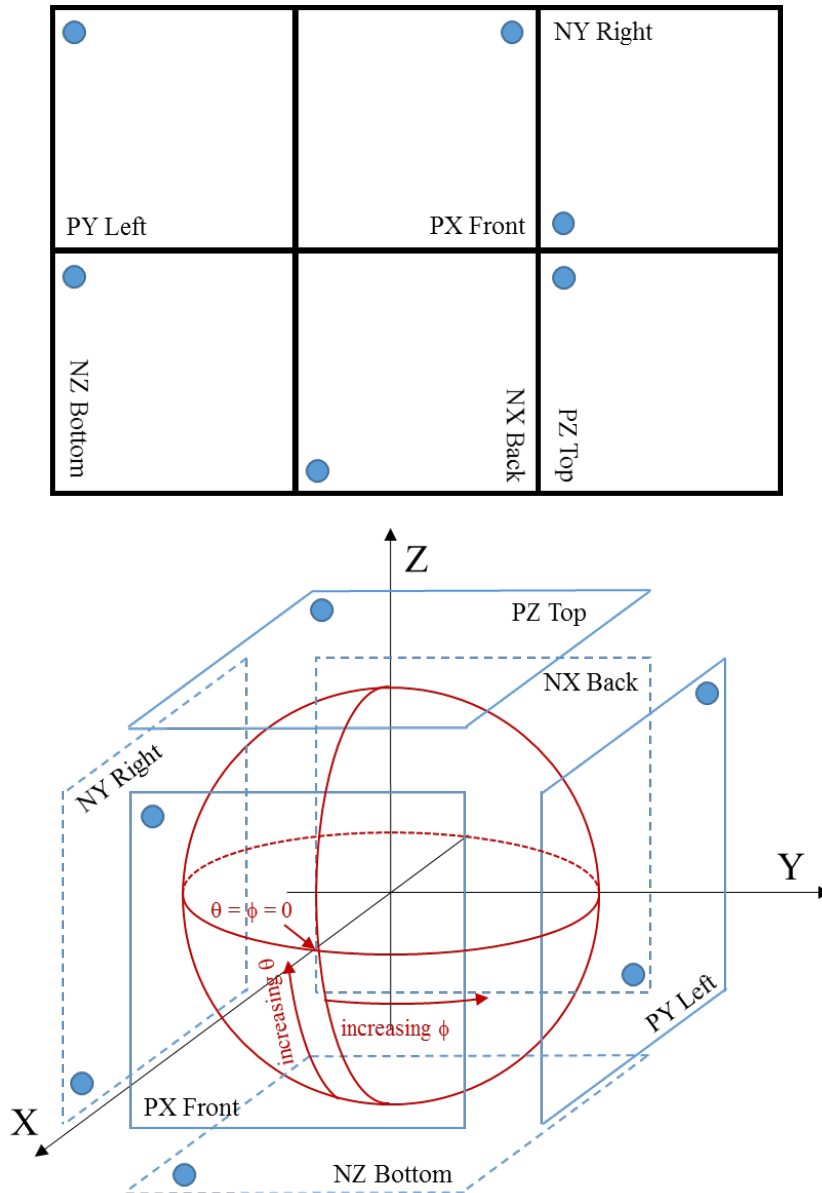


Figure 7 — Relation of the cube face arrangement of the projected picture to the sphere coordinates

The values of pictureWidth and pictureHeight shall be such that pictureWidth is a multiple of 3, pictureHeight is a multiple of 2, and pictureWidth / 3 is equal to pictureHeight / 2.

The sphere coordinates (ϕ , θ) for the luma sample location, in degrees, are given by the following equations:

```

lw = pictureWidth / 3
lh = pictureHeight / 2
tmpHorVal = hPos - Floor( hPos ÷ lw ) * lw
tmpVerVal = vPos - Floor( vPos ÷ lh ) * lh
hPos' = -( 2 * tmpHorVal ÷ lw ) + 1
vPos' = -( 2 * tmpVerVal ÷ lh ) + 1
w = Floor( hPos ÷ lw )
h = Floor( vPos ÷ lh )
if( w == 1 && h == 0 ) { // positive x front face
    x = 1.0
    y = hPos'
    z = vPos'

```

```

} else if( w == 1 && h == 1 ) { // negative x back face
    x = -1.0
    y = -vPos'
    z = -hPos'
} else if( w == 2 && h == 1 ) { // positive z top face
    x = -hPos'
    y = -vPos'
    z = 1.0
} else if( w == 0 && h == 1 ) { // negative z bottom face
    x = hPos'
    y = -vPos'
    z = -1.0
} else if( w == 0 && h == 0 ) { // positive y left face
    x = -hPos'
    y = 1.0
    z = vPos'
} else { // ( w == 2 && h == 0 ), negative y right face
    x = hPos'
    y = -1.0
    z = vPos'
}
}
 $\phi = \text{Atan2}(y, x) * 180 \div \pi$ 
 $\theta = \text{Asin}\left(z \div \sqrt{x^2 + y^2 + z^2}\right) * 180 \div \pi$ 

```

(6)

5.3 Conversion from the local coordinate axes to the global coordinate axes

Inputs to this process are:

- `rotation_yaw` (α_d), `rotation_pitch` (β_d), `rotation_roll` (γ_d), all in units of degrees, where `rotation_yaw` (α_d) and `rotation_roll` (γ_d), are in the range of -180.0, inclusive, to 180.0, exclusive, and `rotation_pitch` (β_d) is in the range of -90.0 to 90.0, inclusive, and
- sphere coordinates (ϕ_d, θ_d) relative to the local coordinate axes.

Outputs of this process are:

- sphere coordinates (ϕ', θ') in degrees relative to the global coordinate axes.

This process specifies rotations around the three axes of the coordinate system of subclause 5.1, where yaw (α_d) expresses a rotation around the Z axis, pitch (β_d) rotates around the Y axis, and roll (γ_d) rotates around the X axis. Rotations are extrinsic, i.e. around X, Y, and Z fixed reference axes. The angles increase clockwise when looking from the origin towards the positive end of an axis, as illustrated in Figure 8.

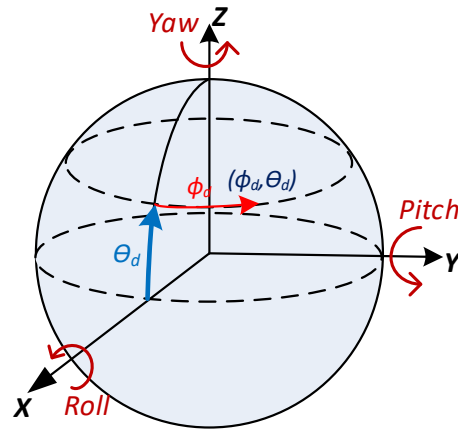


Figure 8 — Illustration of the directions of the yaw, pitch, and roll rotations

When any of the yaw (α_d), pitch (β_d) and roll (γ_d) rotation angles is not equal to zero, an OMAF player needs to apply the sphere rotation process specified in this subclause to convert the local coordinate axes to the global coordinate axes.

It is assumed that the global coordinate systems for different media types were made aligned during content production.

The outputs are derived as follows:

$$\begin{aligned}
 \phi &= \phi_d * \pi \div 180 \\
 \theta &= \theta_d * \pi \div 180 \\
 \alpha &= \alpha_d * \pi \div 180 \\
 \beta &= \beta_d * \pi \div 180 \\
 \gamma &= \gamma_d * \pi \div 180 \\
 x_1 &= \text{Cos}(\phi) * \text{Cos}(\theta) \\
 y_1 &= \text{Sin}(\phi) * \text{Cos}(\theta) \\
 z_1 &= \text{Sin}(\theta) \\
 x_2 &= \text{Cos}(\beta) * \text{Cos}(\alpha) * x_1 - \text{Cos}(\beta) * \text{Sin}(\alpha) * y_1 + \text{Sin}(\beta) * z_1 \\
 y_2 &= (\text{Cos}(\gamma) * \text{Sin}(\alpha) + \text{Sin}(\gamma) * \text{Sin}(\beta) * \text{Cos}(\alpha)) * x_1 + \\
 &\quad (\text{Cos}(\gamma) * \text{Cos}(\alpha) - \text{Sin}(\gamma) * \text{Sin}(\beta) * \text{Sin}(\alpha)) * y_1 - \\
 &\quad \text{Sin}(\gamma) * \text{Cos}(\beta) * z_1 \\
 z_2 &= (\text{Sin}(\gamma) * \text{Sin}(\alpha) - \text{Cos}(\gamma) * \text{Sin}(\beta) * \text{Cos}(\alpha)) * x_1 + \\
 &\quad (\text{Sin}(\gamma) * \text{Cos}(\alpha) + \text{Cos}(\gamma) * \text{Sin}(\beta) * \text{Sin}(\alpha)) * y_1 + \\
 &\quad \text{Cos}(\gamma) * \text{Cos}(\beta) * z_1 \\
 \phi' &= \text{Atan2}(y_2, x_2) * 180 \div \pi \\
 \theta' &= \text{Asin}(z_2) * 180 \div \pi
 \end{aligned} \tag{7}$$

5.4 Region-wise packing formats

5.4.1 General

Subclause 5.4 specifies the inverse processes of the region-wise packing for remapping of a luma sample location in a packed region onto a luma sample location of the corresponding projected region.

The inverse region-wise packing processes specified in this document are identified by a 4-bit unsigned integer value. Table 11 specifies the region-wise packing format identifier values and provides references to the subclauses in which the respective inverse processes are specified.

Table 11 — Region-wise packing formats

Identifier value	Region-wise packing format	Subclause
0	Rectangular region-wise packing	5.4.2
1..15	Reserved	N/A

5.4.2 Conversion of one sample location for rectangular region-wise packing

This subclause specifies the inverse of the rectangular region-wise packing process for remapping of a luma sample location in a packed region onto a luma sample location of the corresponding projected region.

Inputs to this process are:

- sample location (x, y) within the packed region, where x and y are in relative packed picture sample units, while the sample location is at an integer sample location within the packed picture,
- the width and the height (projRegWidth, projRegHeight) of the projected region, in relative projected picture sample units,
- the width and the height (packedRegWidth, packedRegHeight) of the packed region, in relative packed picture sample units,
- transform type (transformType), and
- offset values for the sampling position (offsetX, offsetY) in the range of 0, inclusive, to 1, exclusive, in horizontal and vertical relative packed picture sample units, respectively.

NOTE offsetX and offsetY both equal to 0.5 indicate a sampling position that is in the centre point of a sample in packed picture sample units.

Outputs of this process are:

- the centre point of the sample location (hPos, vPos) within the projected region, where hPos and vPos are in relative projected picture sample units and may have non-integer real values.

The outputs are derived as follows:

```

if( transformType == 0 || transformType == 1 || transformType == 2 || transformType == 3 ) {
    horRatio = projRegWidth ÷ packedRegWidth
    verRatio = projRegHeight ÷ packedRegHeight
} else if ( transformType == 4 || transformType == 5 || transformType == 6 ||
transformType == 7 ) {
    horRatio = projRegWidth ÷ packedRegHeight
    verRatio = projRegHeight ÷ packedRegWidth
}
if( transformType == 0 ) {
    hPos = horRatio * ( x + offsetX )
    vPos = verRatio * ( y + offsetY )
} else if ( transformType == 1 ) {
    hPos = horRatio * ( packedRegWidth - x - offsetX )
    vPos = verRatio * ( y + offsetY )
} else if ( transformType == 2 ) {
    hPos = horRatio * ( packedRegWidth - x - offsetX )
    vPos = verRatio * ( packedRegHeight - y - offsetY )
} else if ( transformType == 3 ) {

```

(8)

```

        hPos = horRatio * ( x + offsetX )
        vPos = verRatio * ( packedRegHeight - y - offsetY )
    } else if ( transformType == 4 ) {
        hPos = horRatio * ( y + offsetY )
        vPos = verRatio * ( x + offsetX )
    } else if ( transformType == 5 ) {
        hPos = horRatio * ( y + offsetY )
        vPos = verRatio * ( packedRegWidth - x - offsetX )
    } else if ( transformType == 6 ) {
        hPos = horRatio * ( packedRegHeight - y - offsetY )
        vPos = verRatio * ( packedRegWidth - x - offsetX )
    } else if ( transformType == 7 ) {
        hPos = horRatio * ( packedRegHeight - y - offsetY )
        vPos = verRatio * ( x + offsetX )
    }

```

6 Fisheye omnidirectional video

6.1 General

Without the *projection* and *region-wise packing* processes specified in Clause 5, multiple circular images captured by fisheye cameras may be directly projected onto a picture, which consists of fisheye omnidirectional video. In an OMAF player, the decoded fisheye omnidirectional video may be stitched and rendered according to the user's intended viewport using the signalled fisheye video parameters, including:

- Region information of circular images in the coded picture,
- Field of view and camera parameters of fisheye lens,
- Lens distortion correction (LDC) parameters with local variation of FOV, and
- Lens shading compensation (LSC) parameters with RGB gains.

Essential fisheye video parameters for enabling stitching and rendering at the OMAF player are specified in `FisheyeVideoEssentialInfoStruct()` in subclause 6.2. For high quality rendering and efficient delivery of fisheye video, supplemental fisheye video parameters are specified in `FisheyeVideoSupplementalInfoStruct()` in subclause 6.3.

6.2 The `FisheyeVideoEssentialInfoStruct()` syntax structure

6.2.1 Syntax

```

aligned(8) class FisheyeVideoEssentialInfoStruct() {
    unsigned int(3) view_dimension_idc;
    bit(21) reserved = 0;
    unsigned int(8) num_circular_images;
    for (i=0; i< num_circular_images; i++) {
        unsigned int(32) circular_image_centre_x;
        unsigned int(32) circular_image_centre_y;
        unsigned int(32) rect_region_top;
        unsigned int(32) rect_region_left;
        unsigned int(32) rect_region_width;
        unsigned int(32) rect_region_height;
    }
}

```