INTERNATIONAL STANDARD

ISO/IEC 23090-2

First edition
2019-01

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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

Reference number
ISO/IEC 23090-2:2019(E)

© ISO/IEC 2019

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2019

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
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2019 – All rights reserved

Contents

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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).

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).

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.

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

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

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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

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 most popular forms of omnidirectional media applications is omnidirectional video, also known as 360° video. Omnidirectional video is typically captured by multiple cameras that cover up to 360° of the scene. 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):

1) 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,

2) 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,

3) storage of omnidirectional media and the associated metadata using the ISO base media file format (ISOBMFF) as specified in ISO/IEC 14496-12,

4) encapsulation, signalling, and streaming of omnidirectional media 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, and

5) media profiles and presentation profiles that provide interoperable and 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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Information technology — Coded representation of immersive media —

Part2:
Omnidirectional media format

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

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.

In an OMAF player the user's viewing perspective is from the centre of the sphere looking outward towards the inside surface of the sphere.

NOTE 1 In this document, only 3 degrees of freedom (3DOF) is supported. In other words, purely translational movement of the user does not result in different omnidirectional media being rendered to the user. For 3DOF support with stereoscopic video, when the user rolls his/her head, there could be a stereoscopic rendering issue.

NOTE 2 Omnidirectional video could contain graphics elements generated by computer graphics but encoded as 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, *Infomation technology — Coding of audio-visual objects — Part 1: Systems*

ISO/IEC 14496-3:2009, *Information technology — Coding of audio-visual objects — Part 3: Audio*

ISO/IEC 14496-10:2014, *Information technology — Coding of audio-visual objects — Part 10: Advanced video coding*

ISO/IEC 14496-12, *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 14496-15:2017, *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 23000-19:2018, *Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media*

ISO/IEC 23003-4:2015, *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)*

ISO/IEC 23008-2:2017, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High efficiency video coding*

ISO/IEC 23008-3:2015, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 3: 3D audio*

© ISO/IEC 2019 – All rights reserved

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

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

ISO/IEC 23091-2[1], *Information technology — Coding-independent code points — Part 2: Video*

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

W3C Recommendation, *TTML profiles for Internet media subtitles and captions 1.0 (IMSC1)*

WebVTT: *The web video text tracks format, W3C* (Working Draft, 08 August 2017)

W3C Recommendation, *XML schema part 1: Structures*

W3C Recommendation, *XML schema part 2: Datatypes*

IETF BCP 47, *Tags for Identifying Languages*

IETF RFC 6381, *MIME Codecs and Profiles*

3 Terms, definitions, abbreviated terms, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions in ISO/IEC 14496-12, ISO/IEC 23008-12, ISO/IEC 23009-1 and the following apply. If terms defined in ISO/IEC 14496-12, ISO/IEC 23008-12 and ISO/IEC 23009-1 are also defined in this document, the definitions in this document are applicable.

NOTE In particular, the terms coded image, coded image item, derived image, derived image item, image item, reconstructed image, and source image item are defined in ISO/IEC 23008-12.

The terminological databases for use in standardization maintained by ISO and IEC at the following addresses:

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

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* (3.1.17).

3.1.3
circular image
image captured with a *fisheye lens* (3.1.14)

3.1.4
closed scheme type
scheme type (3.1.35) that clearly specifies which transformations are allowed and does not allow future extensions

[1] Under preparation. Stage at time of FDIS ballot: ISO/IEC DIS 23091-2, 40.99.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

3.1.5
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

3.1.6
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

3.1.7
content coverage
one or more sphere regions that are covered by the content represented by the track or by an image item

3.1.8
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.

3.1.9
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* (3.1.17). This coincides with the equator on Earth.

3.1.10
extractor track
track that has *untransformed sample entry type* (3.1.44) equal to `'hvc2'`, `'avc2'`, or `'avc4'` and contains one or more `'scal'` track references

3.1.11
field of view
extent of the observable world in captured/recorded content or in a physical display device

3.1.12
file decoder
collective term for file/segment decapsulation and decoding of video, audio or image bitstreams

3.1.13
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 of the following:
– decoded pictures or audio samples, and rendering metadata for them;
– a fully rendered audio scene in the reference system

3.1.14

fisheye lens
wide-angle camera lens that usually captures an approximately hemispherical *field of view* (3.1.11) and projects it as a *circular image* (3.1.3)

3.1.15
fisheye video
video captured by *fisheye lenses* (3.1.14)

3.1.16
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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

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.

3.1.17
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.

3.1.18
guard band
area in a *packed picture* (3.1.23) that is not rendered but may be used to improve the rendered part of the packed picture to avoid or mitigate visual artifacts such as seams

Note to entry: Guard bands are associated with *packed regions* (3.1.24) as described in subclause 7.5.3.

3.1.19
local coordinate axes
coordinate axes obtained after applying rotation to the *global coordinate axes* (3.1.16)

3.1.20
OMAF player
collective term for
– file/segment reception or file access;
– file/segment decapsulation;
– decoding of audio, video, image, or timed text bitstreams;
– rendering of audio, images, or timed text; and
– viewport selection

3.1.21
omnidirectional media
media such as image or video and its associated audio that enable rendering according to the user's *viewing orientation* (3.1.45), if consumed with a head-mounted device, or according to user's desired *viewport* (3.1.46), otherwise, as if the user was in the spot where and when the media was captured

3.1.22
open-ended scheme type
scheme type (3.1.35) that allows future extensions

3.1.23
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* (3.1.31) of a *projected picture* (3.1.25).

3.1.24
packed region
region in a *packed picture* (3.1.24) that is mapped to a *projected region* (3.1.26) as specified by the *region-wise packing* (3.1.31) signalling

3.1.25
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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

3.1.26
projected region
region in a *projected picture* (3.1.25) that is mapped to a *packed region* (3.1.24) as specified by the region-wise packing signalling

3.1.27
projection
inverse of the process by which the samples of a *projected picture* (3.1.25) are mapped to a set of positions identified by a set of azimuth and elevation coordinates on a unit sphere

3.1.28
quality ranking region
region that is associated with a quality ranking value and is specified relative to a decoded picture or a sphere

3.1.29
quality ranking 2D region
quality ranking region (3.1.28) that is specified relative to a decoded picture

3.1.30
quality ranking sphere region
quality ranking region (3.1.28) that is specified relative to a sphere

3.1.31
region-wise packing
inverse of the process of transformation, resizing, and relocating of *packed regions* (3.1.24) of a *packed picture* (3.1.23) to remap to *projected regions* (3.1.26) of a *projected picture* (3.1.25)

3.1.32
rendering
process of generating audio-visual content for playback from the decoded audio-visual data according to the user's *viewing orientation* (3.1.45), if consumed with a head-mounted device, or according to user's desired *viewport* (3.1.46), otherwise

3.1.33
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.

3.1.34
sample entry type
four-character code that is either the value of the `format` field of a `SampleEntry` directly contained in `SampleDescriptionBox` or the `data_format` value of an instance of `OriginalFormatBox`

3.1.35
scheme type
type that parameterizes an encrypted, restricted, or incomplete media track

3.1.36
sphere coordinates
azimuth (ϕ) and *elevation* (θ) that identify a location of a point on the unit sphere

3.1.37
sphere region
region on a sphere, specified either by four *great circles* (3.1.17) or by two *azimuth circles* (3.1.2) and two *elevation circles* (3.1.9), or such a region on the rotated sphere after applying certain amount of yaw, pitch, and roll rotations

3.1.38
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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

3.1.39
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

3.1.40
sub-picture track
track that represents a *sub-picture bitstream* (3.1.39)

3.1.41
tilt angle
angle indicating the amount of tilt of a *sphere region* (3.1.37), 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

3.1.42
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

3.1.43
track sample entry type
sample entry type (3.1.34) of a `SampleEntry` directly contained in `SampleDescriptionBox`

3.1.44
untransformed sample entry type
track sample entry type (3.1.43) that would apply if no transformations had been performed to a transformed media track

3.1.45
viewing orientation
triplet of azimuth, elevation, and tilt angle characterizing the orientation that a user is consuming the audio-visual content; in case of image or video, characterizing the orientation of the *viewport* (3.1.46)

3.1.46
viewport
region of omnidirectional image or video suitable for display and viewing by the user

3.2 Abbreviated terms

2D	two-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)
DASH	MPEG dynamic adaptive streaming over HTTP (specified in ISO/IEC 23009-1)
ERP	equirectangular projection
FOV	field of view
ISOBMFF	ISO base media file format (specified in ISO/IEC 14496-12)
HEVC	high efficiency video coding (specified in ISO/IEC 23008-2)
HMD	head-mounted display
HOA	higher-order ambisonics

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

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

3.3 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 EXAMPLE 7 / 4 and −7 / −4 are truncated to 1 and −7 / 4 and 7 / −4 are truncated to −1.
÷	denotes division in mathematical equations where no truncation or rounding is intended
$\frac{x}{y}$	denotes 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 >= 0 and y > 0.)
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 −π÷2 to π÷2, inclusive, in units of radians
Atan(x)	trigonometric invers tangent function, operating on an argument x that is any real number, with an output value in the range of −π÷2 to π÷2, inclusive, in units of radians

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

Cos(x)	trigonometric cosine function operating on an argument x in units of radians

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Floor(x) largest integer less than or equal to x

Round(x) = Sign(x) * Floor(Abs(x) + 0.5)

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

Sin(x) trigonometric sine function operating on an argument x in units of radians

Tan(x) trigonometric tangent function operating on an argument x in units of radians

3.4 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"

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

3.5 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

3.6 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 clause in which they are derived.

3.7 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.

4 Overview

4.1 Organization of this document

This document is organized as follows:

- Subclauses 4.2 and 4.3 provide the overall architecture for projected omnidirectional video and fisheye omnidirectional video, respectively.
- Subclause 4.4 describes which types of referenceable features specified in this document implementations and external specifications may be used to achieve interoperability. It also defines OMAF media and presentation profiles.
- Clause 5 specifies a coordinate system used in this document and the equations for the equirectangular and cubemap omnidirectional projection formats, the conversion from the local coordinate axes to the global coordinate axes, and the rectangular region-wise packing.
- Clause 6 specifies syntax structures that are common for fisheye video and fisheye images.
- Clause 7 specifies extensions to the ISO base media file format for omnidirectional media as well as for timed metadata for sphere regions. It also specifies generic extensions to ISO/IEC 14496-12 and ISO/IEC 14496-15, which may be used also for other purposes than for omnidirectional media.
- Clause 8 specifies extensions to DASH for omnidirectional media.
- Clause 9 specifies extensions to MMT for omnidirectional media.
- Clause 10 specifies OMAF media profiles.
- Clause 11 specifies OMAF presentation profiles based on some of the OMAF media profiles specified in Clause 10.
- Annex A contains the OMAF DASH XML schema.
- Annex B specifies the DASH integration of all the OMAF media profiles for timed media specified in Clause 10.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- Annex C specifies the CMAF integration of some of the OMAF media profiles specified in Clause 10.
- Annex D describes some informative schemes for viewport-dependent omnidirectional video processing.
- Annex E contains some informative DASH MPD examples.
- Annex F contains some informative MMT signalling examples.

4.2 Overall architecture for omnidirectional media with projected video

4.2.1 Overview

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

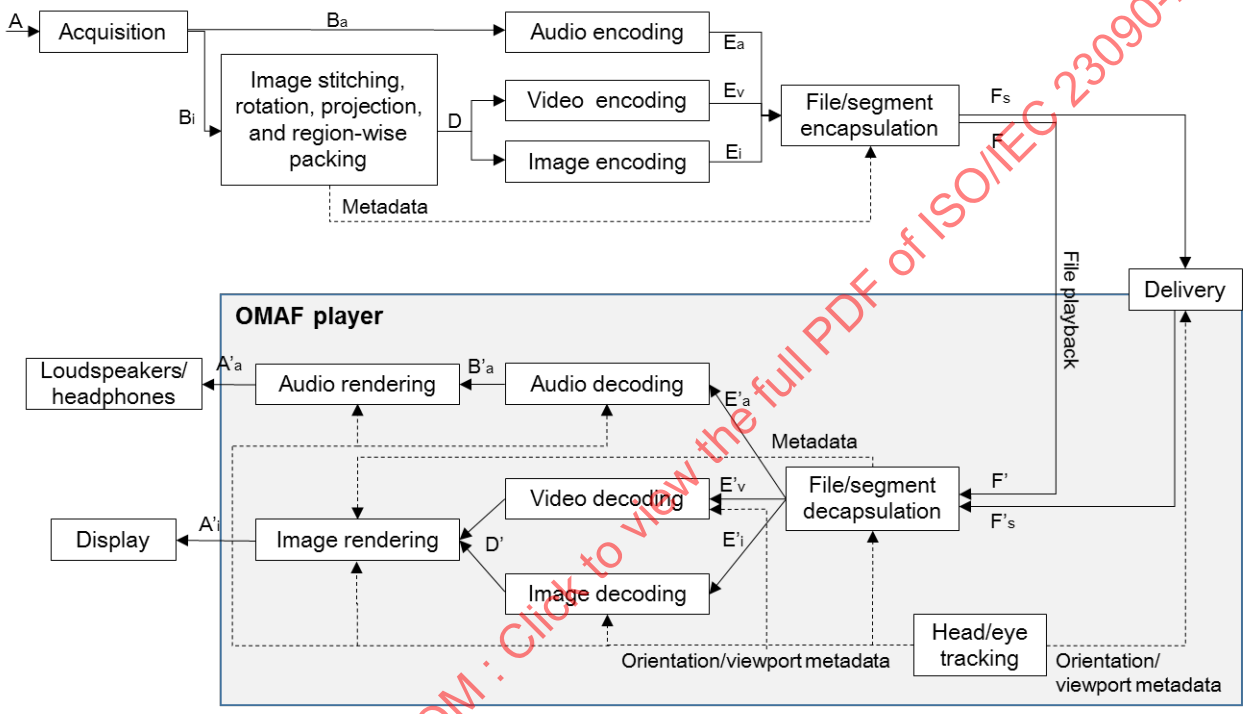


Figure 1 – Content flow process for omnidirectional media with projected video

The following interfaces are normatively 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, including projection and region-wise packing metadata; 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 normatively in this document.

NOTE While the syntax and semantics of the bitstreams Ea, Ev, and Ei are the same as those for E'a, E'v, E'i, respectively, the input interface to the file/segment encapsulation module is not normatively 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-degree video.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

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 binaraulized for presentation via headphones.

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.

For audio, the stitching process is not needed, since the captured signals are inherently immersive and omnidirectional.

The packed 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, and/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, such as projection and region-wise packing information assisting in rendering the decoded packed pictures.

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, and

– region-wise quality ranking.

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, and/or E'_i) and parses the metadata. The audio, video, and/or images are then decoded into decoded signals (B'_a for audio, and D' for images/video). 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. 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 head tracking and possibly also eye tracking functionality. 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.

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

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

For monoscopic 360-degree 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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

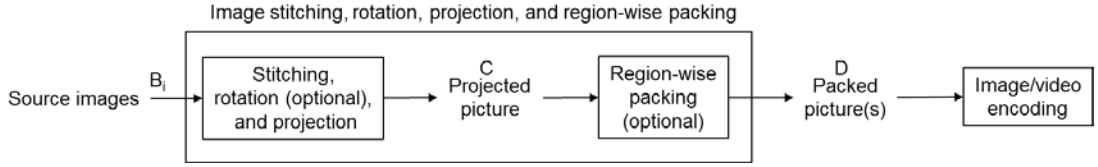


Figure 2 – Monoscopic image stitching, rotation, projection, and region-wise packing of source images of a single acquisition time instance

In the case of stereoscopic 360-degree 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.

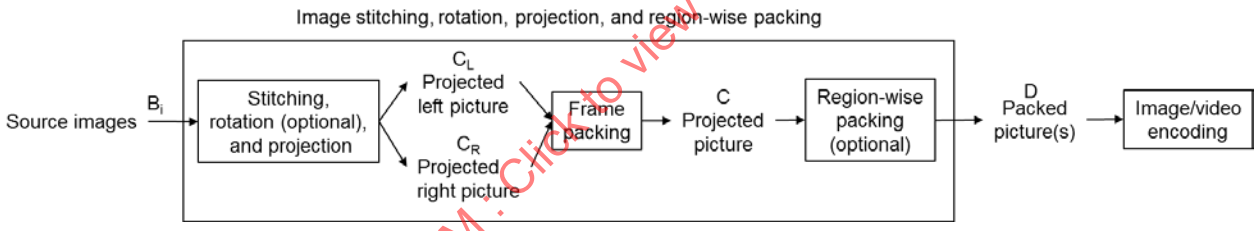


Figure 3 – Stereoscopic image stitching, rotation, projection, and region-wise packing of source images of a single acquisition time instance

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 Overall architecture for omnidirectional media with fisheye video

Figure 4 shows a typical content flow process for an omnidirectional media application with fisheye video.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

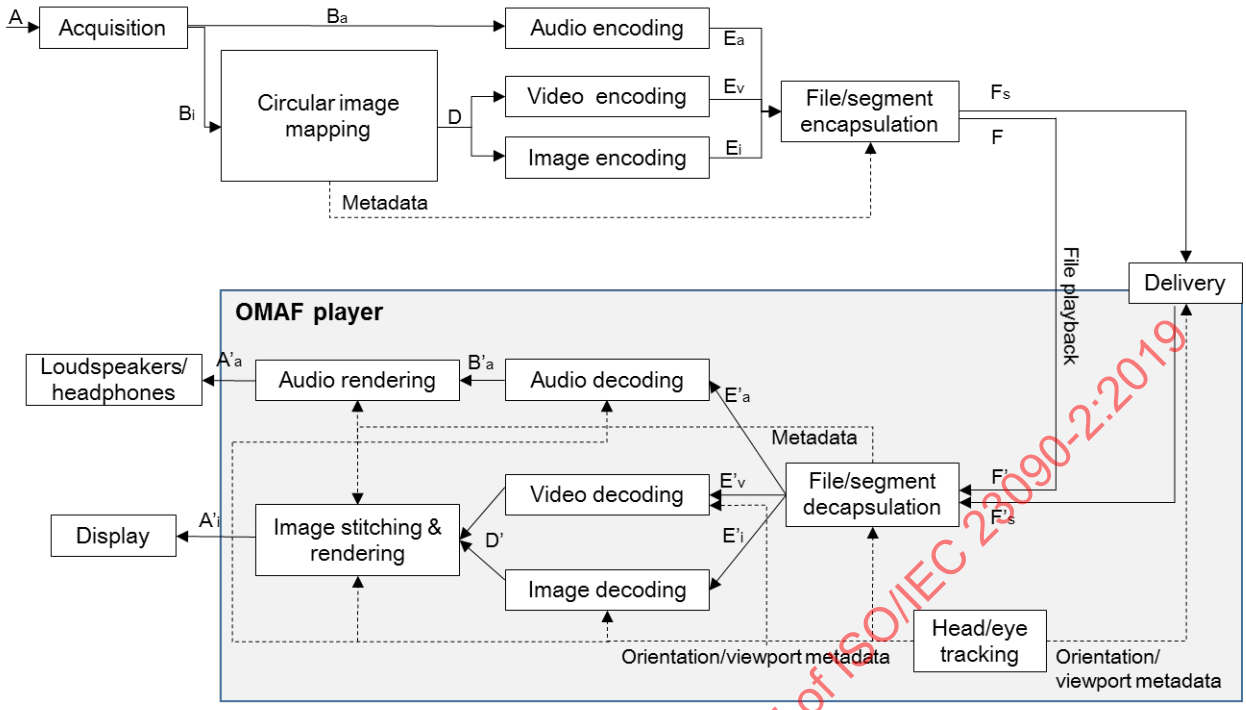


Figure 4 – Content flow process for omnidirectional media with fisheye video

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, and/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 Conformance and interoperability

4.4.1 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).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- Omnidirectional projection formats (subclause 5.2).
- Media profiles, as defined in subclause 4.4.2. Individual media profiles are specified in Clause 10.
- Presentation profiles, as defined in subclause 4.4.3. 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.
- 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.

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 4.4.4.

4.4.2 Media profiles

4.4.2.1 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 could 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.

4.4.2.2 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,

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

- 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.

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 5 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. While the player operation, with the exception of the file decoding process, is not specified normatively in this document, specifications of a media profile may include an informative clause on expectations of a player operation, for example including recommendations for rendering.

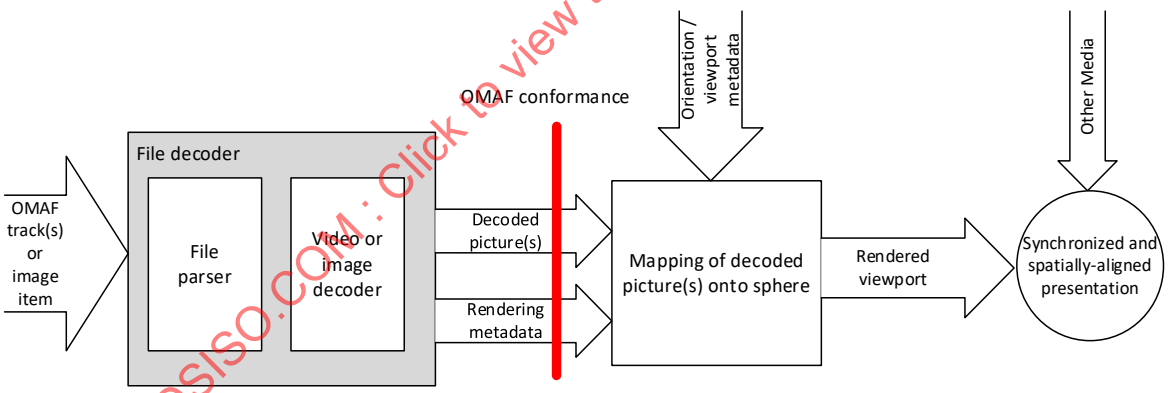


Figure 5 – Conformance points for OMAF video and image media profiles

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

4.4.2.3 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.

4.4.3 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.

© ISO/IEC 2019 – All rights reserved

4.4.4 Summary of referenceable code points

4.4.4.1 Brands

ISO/IEC 14496-12 defines the concept of brands, which may be indicated in the `FileTypeBox`. This document specifies the `TrackTypeBox`, which may be used to indicate brands in a track-specific manner. Brands are used in this document to indicate conformance to OMAF media profiles, OMAF presentation profiles, 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	Clause 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
`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
`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
`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
`cabl`	C.2.1	CMAF media profile for OMAF 3D audio baseline profile

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

4.4.4.2 Uniform resource names

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

© ISO/IEC 2019 – All rights reserved

Table 3 – URNs specified in this document

URN	Clause in this document	Informative description
urn:mpeg:mpegI:omaf:2017	8.2.2	Namespace for the XML elements and attributes specified in this document
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

4.4.4.3 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	Clause in this document	Informative 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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

4.4.4.4 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	Clause in this document	Informative 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.
`ttsl`	7.7.6	For use with a timed metadata track containing the timed text sphere location timed metadata.

4.4.4.5 Box types

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

Table 6 – Box types specified in this document

Box type	Clause in this document	Box name
`csch`	7.1.6	Compatible scheme type box
`ttyp`	7.1.9	Track type box
`povd`	7.6.2	Projected omnidirectional video box
`prfr`	7.6.2, 7.9.3	Projection format box, projection format item property
`fodv`	7.6.3	Fisheye omnidirectional video box
`fovi`	7.6.3, 7.9.4	Fisheye video essential information box, essential fisheye image item property
`fvsi`	7.6.3, 7.9.5	Fisheye video supplemental information box, supplemental fisheye image item property
`rwpk`	7.6.4, 7.9.6	Region-wise packing box, region-wise packing item property
`rotn`	7.6.5, 7.9.7	Rotation box, rotation item property
`covi`	7.6.6, 7.9.8	Coverage information box, coverage information item property
`rosc`	7.7.2	Sphere region configuration box
`rvif`	7.7.5	Recommended viewport information box
`srqr`	7.8.2	Sphere region quality ranking box

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Box type	Clause in this document	Box name
2dqr	7.8.3	2D region quality ranking box
stvi	7.9.2	Frame packing item property NOTE FramePackingProperty has the same box type and syntax as StereoVideoBox specified in ISO/IEC 14496-12.
iivo	7.9.9	Initial viewing orientation item property
otcf	7.10.2	OMAF timed text configuration box

4.4.4.6 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	Clause in this document	Track grouping name
ster	7.1.1	Stereoscopic video track grouping
alte	7.2.1	Alternative extraction source track grouping

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

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

© ISO/IEC 2019 – All rights reserved

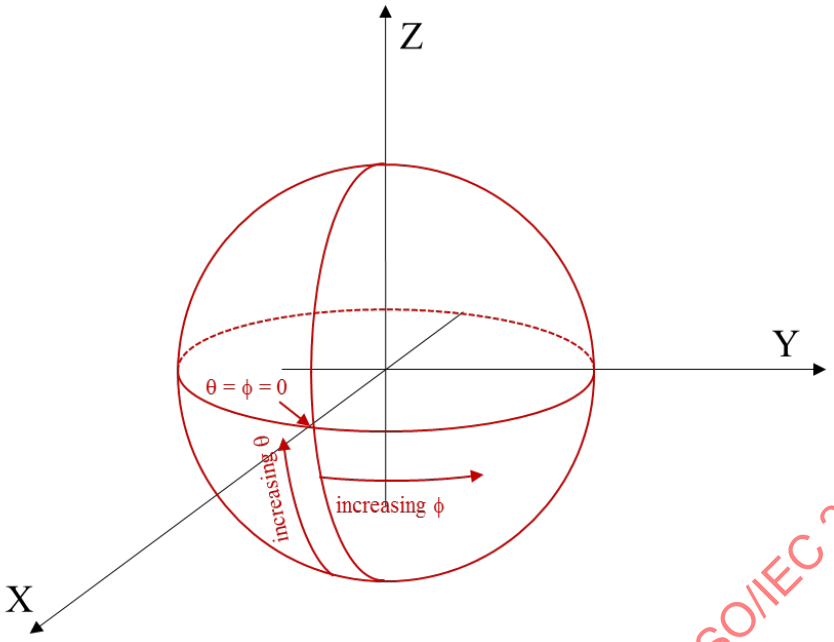


Figure 6 – Coordinate axes and their relation to the sphere coordinates

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

The value ranges of azimuth is −180.0, inclusive, to 180.0, exclusive, degrees. The value range of elevation is −90.0 to 90.0, inclusive, degrees.

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 8 specifies the omnidirectional projection format identifier values and provides references to the clauses in which the respective inverse projection processes are specified.

Table 8 – Omnidirectional projection formats

Identifier value	Omnidirectional projection	Clause
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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

– 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:

$$\phi = (0.5 - hPos \div pictureWidth) * 360$$

$$\theta = (0.5 - vPos \div pictureHeight) * 180$$

Figure 7 illustrates the azimuth and elevation ranges of a monoscopic projected picture with the equirectangular projection.

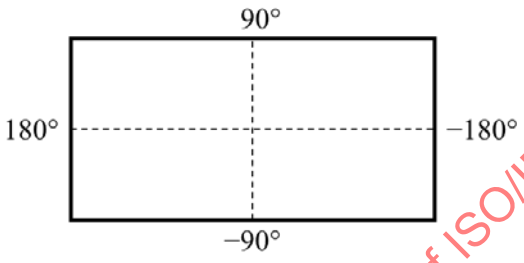


Figure 7 – Azimuth and elevation ranges of the monoscopic projected picture of the equirectangular projection

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 8 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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

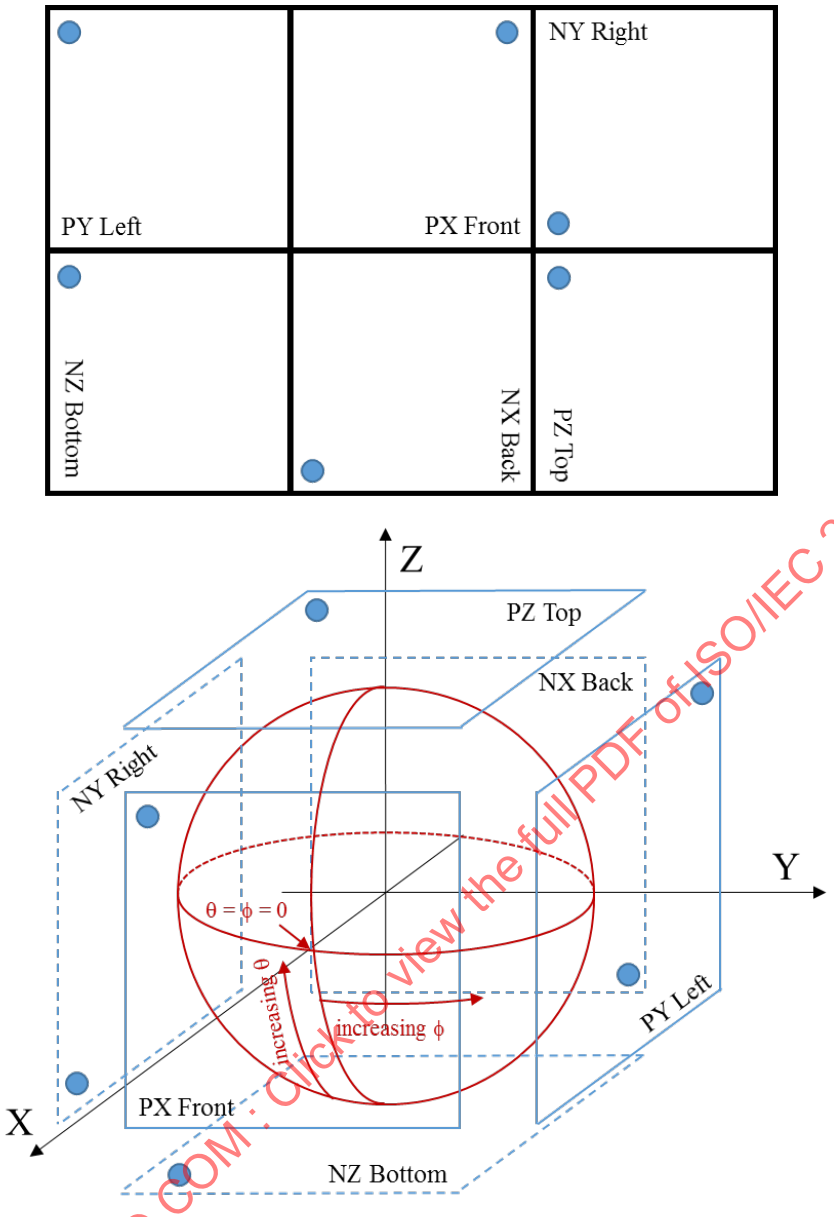


Figure 8 – Relation of the cube face arrangement of the projected picture to the sphere coordinates

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
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
        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′
    }
    ϕ = Atan2( y, x ) * 180 ÷ π
    θ = Asin( z ÷ √(x² + y² + z²) ) * 180 ÷ π
```

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 9.

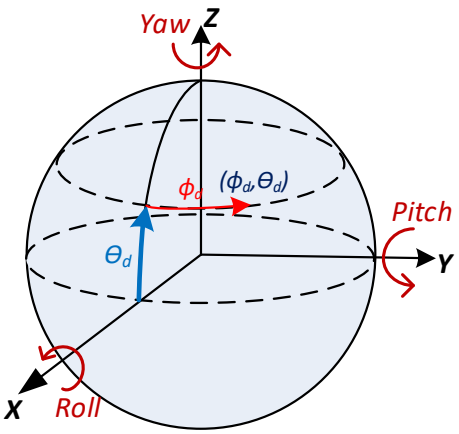


Figure 9 – Illustration of the directions of the yaw, pitch and roll rotations

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

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 clause 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:

$$\phi = \phi_d * \pi \div 180$$
$$\theta = \theta_d * \pi \div 180$$
$$\alpha = \alpha_d * \pi \div 180$$
$$\beta = \beta_d * \pi \div 180$$
$$\gamma = \gamma_d * \pi \div 180$$
$$x_1 = Cos(\phi) * Cos(\theta)$$
$$y_1 = Sin(\phi) * Cos(\theta)$$
$$z_1 = Sin(\theta)$$
$$x_2 = Cos(\beta) * Cos (\alpha) * x_1 - Cos(\beta) * Sin(\alpha) * y_1 + Sin(\beta) * z_1$$

$$y_2 = (Cos(\gamma) * Sin(\alpha) + Sin(\gamma) * Sin(\beta) * Cos(\alpha)) * x_1 + (Cos(\gamma) * Cos(\alpha) - Sin(\gamma) * Sin(\beta) * Sin(\alpha)) * y_1 - Sin(\gamma) * Cos(\beta) * z_1$$

$$z_2 = (Sin(\gamma) * Sin(\alpha) - Cos(\gamma) * Sin(\beta) * Cos(\alpha)) * x_1 + (Sin(\gamma) * Cos(\alpha) + Cos(\gamma) * Sin(\beta) * Sin(\alpha)) * y_1 + Cos(\gamma) * Cos(\beta) * z_1$$

$$\phi' = Atan2(y_2, x_2) * 180 \div \pi$$
$$\theta' = Asin(z_2) * 180 \div \pi$$

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 9 specifies the region-wise packing format identifier values and provides references to the clauses in which the respective inverse processes are specified.

Table 9 – Region-wise packing formats

Identifier value	Region-wise packing format	Clause
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 clause 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,

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

– 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 ) {
    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:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

– 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 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) image_centre_x;
       unsigned int(32) 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;
       unsigned int(32) full_radius;
       unsigned int(32) scene_radius;
       signed int(32) camera_centre_azimuth;
       signed int(32) camera_centre_elevation;
       signed int(32) camera_centre_tilt;
       unsigned int(32) camera_centre_offset_x;
       unsigned int(32) camera_centre_offset_y;
       unsigned int(32) camera_centre_offset_z;
       unsigned int(32) field_of_view;
       bit(16) reserved = 0;
       unsigned int(16) num_polynomial_coefs_distortion;
       for (j=0; j< num_polynomial_coefs_distortion; j++)
           signed int(32) polynomial_coef_k_distortion;
    }
}
```

6.2.2 Semantics

`view_dimension_idc` indicates that the syntax element values of this `FisheyeVideoEssentialInfoStruct` syntax structure have been constrained as specified in Table 10. The indicated constraints shall be obeyed in the syntax element values of this `FisheyeVideoEssentialInfoStruct` syntax structure.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

Table 10 – Constraints implied by view_dimension_idc

view_dimension_idc	Constraints
0	num_circular_images is equal to 2. The values of camera_centre_azimuth, camera_centre_elevation, camera_centre_tilt, camera_centre_offset_x, camera_centre_offset_y, and camera_centre_offset_z are such that the circular images have aligned optical axes and face opposite directions. The sum of field_of_view values is greater than or equal to $360 * 2^{16}$.
1	num_circular_images is equal to 2. The values of camera_centre_azimuth, camera_centre_elevation, camera_centre_tilt, camera_centre_offset_x, camera_centre_offset_y, and camera_centre_offset_z are such that the circular images have parallel optical axes that are orthogonal to the line intersecting the camera centre points. The camera with i equal to 0 is the left view.
2	num_circular_images is equal to 2. The values of camera_centre_azimuth, camera_centre_elevation, camera_centre_tilt, camera_centre_offset_x, camera_centre_offset_y, and camera_centre_offset_z are such that the circular images have parallel optical axes that are orthogonal to the line intersecting the camera centre points. The camera with i equal to 0 is the right view.
3 to 6, inclusive	Reserved.
7	No additional constraints are implied for the syntax element values within this FisheyeVideoEssentialInfoStruct syntax structure.

num_circular_images specifies the number of circular images in each coded picture this structure applies to. Typically, the value is equal to 2, but other non-zero values are also possible.

image_centre_x is a fixed-point 16.16 value that specifies the horizontal coordinate, in luma samples, of the centre of the circular image in each coded picture this structure applies to, relative to the top-left corner of the coded picture.

image_centre_y is a fixed-point 16.16 value that specifies the vertical coordinate, in luma samples, of the centre of the circular image in each coded picture this structure applies to, relative to the top-left corner of the coded picture.

rect_region_top, rect_region_left, rect_region_width, and rect_region_height specify the coordinates of the top-left corner and the width and height of the rectangular region that contains the circular image, which may or may not be cropped. These values are specified in units of luma samples.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

full_radius is a fixed-point 16.16 value that specifies the radius, in luma samples, from the centre of the circular image to the edge of the circular image that corresponds to the maximum field of view of the fisheye lens, specified by field_of_view.

The active pixel area of a circular image is defined by the intersection of the rectangular region, defined by rect_region_top, rect_region_left, rect_region_width, and rect_region_height, and the circle defined by image_centre_x, image_centre_y, and full_radius.

scene_radius is a fixed-point 16.16 value that specifies the radius, in luma samples, from the centre of the circular image to the closest edge of the area in the image where there are no obstructions from the camera body itself, and that the enclosed area is the suggested area for stitching as recommended by the content provider.

camera_centre_azimuth and camera_centre_elevation specify the centre of the sphere region corresponding to the centre of the circular image. camera_centre_azimuth shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. camera_centre_elevation shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive.

camera_centre_tilt specifies the tilt angle of the centre of the sphere region corresponding to the centre of the circular image. camera_centre_tilt shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive.

camera_centre_offset_x, camera_centre_offset_y and camera_centre_offset_z are fixed-point 16.16 values that indicate the XYZ offset values, in millimeters, of the focal centre of the fisheye camera lens corresponding to the circular image, from the focal centre origin of the overall fisheye camera configuration as illustrated in Figure 10.

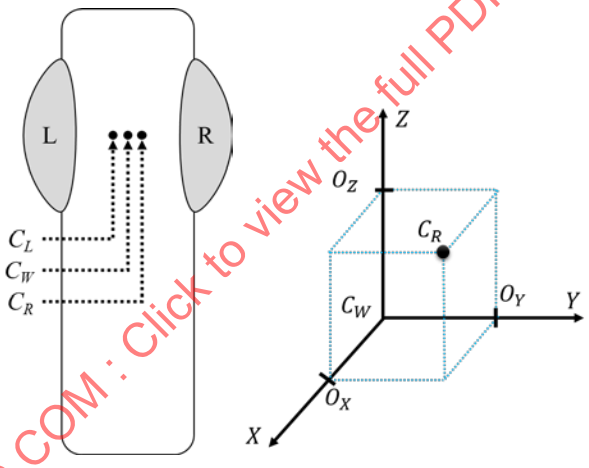


Figure 10 – Illustration of camera_centre_offset_x (O_X), camera_centre_offset_y (O_Y), and camera_centre_offset_z (O_Z) of the focal centre (C_L or C_R) of the fisheye camera lens (L or R, respectively) relative to the focal centre origin of the overall fisheye camera configuration (C_W)

field_of_view is a fixed-point 16.16 value that specifies the field of view of the fisheye lens corresponding to the circular image, in degrees. A typical value for a hemispherical fisheye lens is 180.0 degrees.

num_polynomial_coefs_distortion and polynomial_coef_k_distortion are lens distortion parameters that define the curve function between the normalized radius (r) of luma sample x_p in the circular image and the angle (θ) in the camera sphere where X_p and X_c are the corresponding points of x_p and x_c as shown in Figure 11. The normalized radius (r) of x_p is calculated as the distance (d) divided by full_radius where the distance (d) is the distance between x_p and the centre of the circular image, x_c in units of luma samples.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

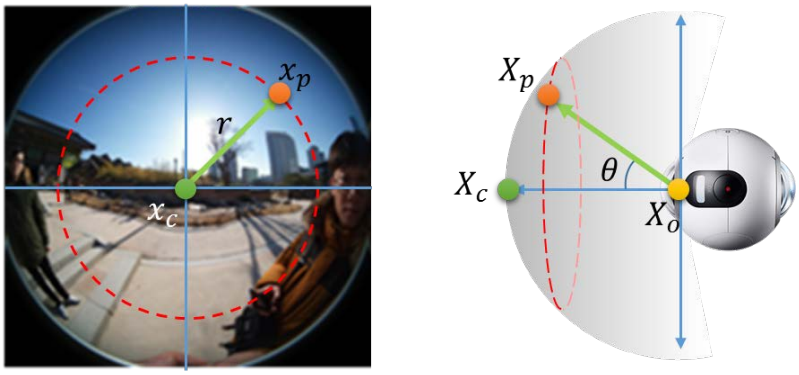


Figure 11 – Illustration of relation between radius (*r*) and theta (*θ*) of distortion function

`num_polynomial_coefs_distortion` is an integer that specifies the number of polynomial coefficients corresponding to the circular image. This is the maximum order of the polynomial plus 1.

The instances of polynomial `polynomial_coef_k_distortion` are fixed-point 8.24 polynomial coefficient values that describes the curve function from the normalized radius (r) of luma sample x_p in the circular image to the angle (θ), in radians, using the following polynomial equation:

$$\theta = \sum_{j=0}^{N_i-1} p_j * r^j$$

where p_j and N_i are represented by `polynomial_coef_k_distortion` and `num_polynomial_coefs_distortion`, respectively.

6.3 FisheyeVideoSupplementalInfoStruct syntax structure

6.3.1 Syntax

```
aligned(8) class FisheyeVideoSupplementalInfoStruct() {
    unsigned int(8) num_circular_images;
    bit(20) reserved = 0;
    unsigned int(1) flip_info_flag;
    unsigned int(1) camera_intrinsic_flag;
    unsigned int(1) local_fov_flag;
    unsigned int(1) deadzone_flag;

    for (i=0; i<num_circular_images; i++) {
       if (flip_info_flag == 1) {
           bit(30) reserved = 0;
           unsigned int(2) image_flip;
       }
       if (camera_intrinsic_flag == 1) {
           unsigned int(32) image_scale_axis_angle;
           unsigned int(32) image_scale_x;
           unsigned int(32) image_scale_y;
           bit(16) reserved = 0;
           unsigned int(16) num_polynomial_coefs_lsc;
           for (j=0; j<num_polynomial_coefs_lsc; j++) {
               signed int (32) polynomial_coef_k_lsc_r;
               signed int (32) polynomial_coef_k_lsc_g;
               signed int (32) polynomial_coef_k_lsc_b;
           }
       }
       if (local_fov_flag == 1) {
           unsigned int (16) num_angle_for_displaying_fov;
           bit(16) reserved = 0;
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

```
            for (j=0; j<num_angle_for_displaying_fov; j++) {
                unsigned int(32) displayed_fov;
                unsigned int(32) overlapped_fov;
            }
            bit(16) reserved = 0;
            unsigned int (16) num_local_fov_region;
            for (j=0; j<num_local_fov_region; j++) {
                unsigned int(32) start_radius;
                unsigned int(32) end_radius;
                signed int(32) start_angle;
                signed int(32) end_angle;
                unsigned int(32) radius_delta;
                signed int(32) angle_delta;
                for (rad=start_radius; rad<=end_radius; rad+=radius_delta)
                    for (ang=start_angle; ang<= end_angle; ang+=angle_delta)
                        unsigned int(32) local_fov_weight;
            }
        }
    }
    if (deadzone_flag == 1) {
        bit(24) reserved = 0;
        unsigned int(8) num_deadzones;
        for (j=0; j<num_deadzones; j++) {
            unsigned int(16) deadzone_left_horizontal_offset;
            unsigned int(16) deadzone_top_vertical_offset;
            unsigned int(16) deadzone_width;
            unsigned int(16) deadzone_height;
        }
    }
}
```

6.3.2 Semantics

`num_circular_images` specifies the number of circular images in each coded picture this structure applies to. Typically, the value is equal to 2, but other non-zero values are also possible.

`flip_info_flag` equal to 1 specifies that `image flipping` region information is present. `flip_info_flag` equal to 0 specifies that image flipping information is not present.

`camera_intrinsic_flag` equal to 1 specifies that image scale and lens shading compensation camera intrinsic parameters are present. `camera_intrinsic_flag` equal to 0 specifies that image scale and lens shading compensation camera intrinsic parameters are not present.

`local_fov_flag` equal to 1 specifies that the local field of view parameters are present. `local_fov_flag` equal to 0 specifies that the local field of view parameters are not present.

`deadzone_flag` equal to 1 specifies that deadone related parameters are present. `deadzone_flag` equal to 0 specifies that deadzone parameters are not present.

`image_flip` specifies whether and how the image has been flipped and thus a reverse flipping operation needs to be applied. The value 0 indicates that the image has not been flipped. The value 1 indicates that the image has been vertically flipped. The value 2 indicates that the image has been horizontally flipped. The value 3 indicates that the image has been both vertically and horizontally flipped.

`image_scale_axis_angle`, `image_scale_x`, and `image_scale_y` are three fixed-point 16.16 values that specify whether and how the circular image has been scaled along an axis. They are used to take into account the natural error in the camera-mirror setting. The axis is defined by a single angle as indicated by the value of `image_scale_axis_angle`, in degrees. An angle of 0 degrees means a horizontal vector is perfectly horizontal and a vertical vector is perfectly vertical. The values of `image_scale_x` and `image_scale_y` indicate the scaling. They may also be called as affine parameters that satisfy the following equation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u_N \\ v_N \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

The equation relates the actual pixel coordinates (u, v) to the ideal image coordinates (u_N, v_N). c_x and c_y represent `image_centre_x`, `image_centre_y` respectively. c, d, and e represent `image_scale_x`, `image_scale_axis_angle`, and `image_scale_y` respectively.

`num_polynomial_coefs_lsc` specifies the number of polynomial coefficients of the lens shading compensation (LSC) parameters for the circular image.

`polynomial_coef_k_lsc_r`, `polynomial_coef_k_lsc_g`, and `polynomial_coef_k_lsc_b` are 8.24 fixed-point values that specify the LSC parameters to compensate the shading artifact of the fisheye lense that reduces the color in the radial direction. The compensating weight (w) to be multiplied to the original color is approximated as a curve function using the following polynomial equation:

$$w = \sum_{i=1}^{N} p_{i-1} \cdot r^{i-1}$$

where r is the normalized radius as specified in subclause 6.2.2.

The weighing factors for red, green, and blue may be calculated separately when p is represented by `polynomial_coef_k_lsc_r`, `polynomial_coef_k_lsc_g`, and `polynomial_coef_k_lsc_b`, respectively, and r is the corresponding radius from the image centre after normalization by the `full_radius`. N is represented by `num_polynomial_coefs_lsc`. Figure 12 shows an example of an input fisheye video picture and the output after applying lens shading compensation.

Figure 12 – Example of a pair of output images (right) resulting from lens shading compensation from a pair of circular input images (left)

`num_angle_for_displaying_fov` specifies the number of angles that define the displayed and overlapped regions. According to the value of `num_angle_for_displaying_fov`, multiple values of `displayed_fov` and `overlapped_fov` are defined with equal intervals, which start at 12 o'clock and increase clockwise.

`displayed_fov` specifies the field of view of the part of the circular image that is recommended to be used for displaying without blending that involves adjacent circular images.

`overlapped_fov` specifies the field of view of the part of the circular image that may be overlapped on a sphere by adjacent circular images and that is recommended to be used for displaying either as such or by blending with adjacent circular images.

The values of `displayed_fov` and `overlapped_fov` shall be less than or equal to the value of `field_of_view`.

NOTE The value of `field_of_view` is determined by the physical property of each fisheye lens, while the values of `displayed_fov` and `overlapped_fov` are determined by the configuration of multiple fisheye lenses. For example, when the value of `num_circular_images` is equal to 2 and two lenses are symmetrically located, the value of `displayed_fov` and `overlapped_fov` could be set as 180 and 190 respectively, by default. However, the value could be changed depending on the configuration of the lens and the characteristics of the contents. For example, if the stitching quality with the `displayed_fov` values (left camera = 170 and right camera = 190) and the `overlapped_fov` values (left camera = 185 and right camera = 190) is better than the quality with the default values (180 and 190) or if the physical configuration of cameras is asymmetric, then the unequal `displayed_fov` and `overlapped_fov` values could be taken. In addition, when it comes to multiple (N>2) fisheye images, a single `displayed_fov` value cannot specify the exact area of each fisheye image. As shown in Figure 13 and Figure 14, `displayed_fov` (red-coloured) varies according to the direction. In order to manipulate multiple (N>2) fisheye images, `num_angle_for_displaying_fov` is introduced. For example, if this value is equal to 12, then the fisheye image is divided into 12 sectors where each sector angle is 30 degrees.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

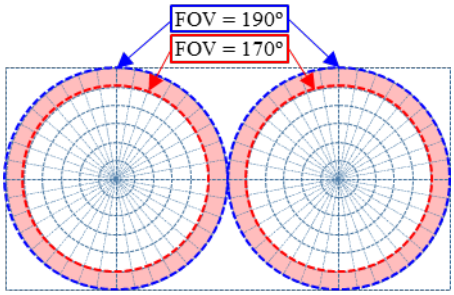


Figure 13 – Displayed FOV for two fisheye images (=170 or 190)

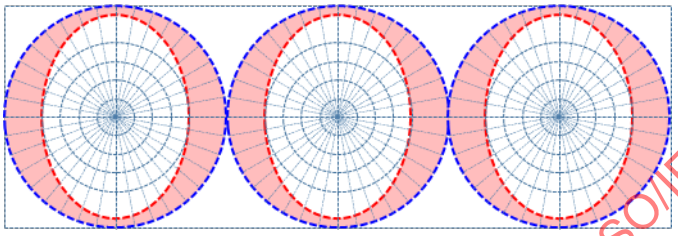


Figure 14 – Displayed FOV and overlapped FOV for multiple (N>2) fisheye images

`num_local_fov_region` specifies the number of local fitting regions that have different fields of view.

`start_radius` (r_s), `end_radius` (r_e), `start_angle` (a_s), and `end_angle` (a_e) specify the region for local fitting/warping to change the actual field of view for displaying locally. `start_radius` and `end_radius` are fixed-point 16.16 values that specify the minimum and maximum radius values. `start_angle` and `end_angle` specify the minimum and maximum angle values that start at 12 o'clock and increase clockwise, in units of 2^{-16} degrees. `start_angle` and `end_angle` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. Figure 15 illustrates an example of local FOV parameters.

`radius_delta` (r_d) is a fixed-point 16.16 value that specifies the delta radius value for representing a different field of view for each radius.

`angle_delta` (a_d) specifies the delta angle value, in units of 2^{-16} degrees, for representing a different field of view for each angle.

`local_fov_weight` (W) is a 8.24 fixed point value which specifies the weighting value for the field of view of the position specified by `start_radius`, `end_radius`, `start_angle`, `end_angle`, the angle index i and the radius index j. The positive value of `local_fov_weight` specifies the expansion of field of view, while the negative value specifies the contraction of field of view. Figure 16 shows an example result of applying local FOV parameters with a set of weighting values.

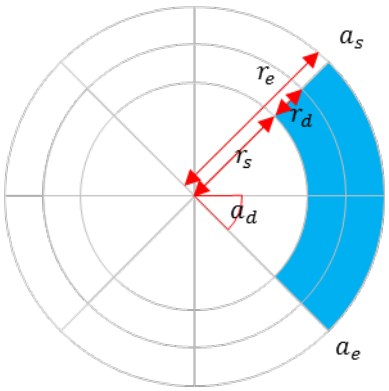


Figure 15 – Illustration of parameters regarding local FOV

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

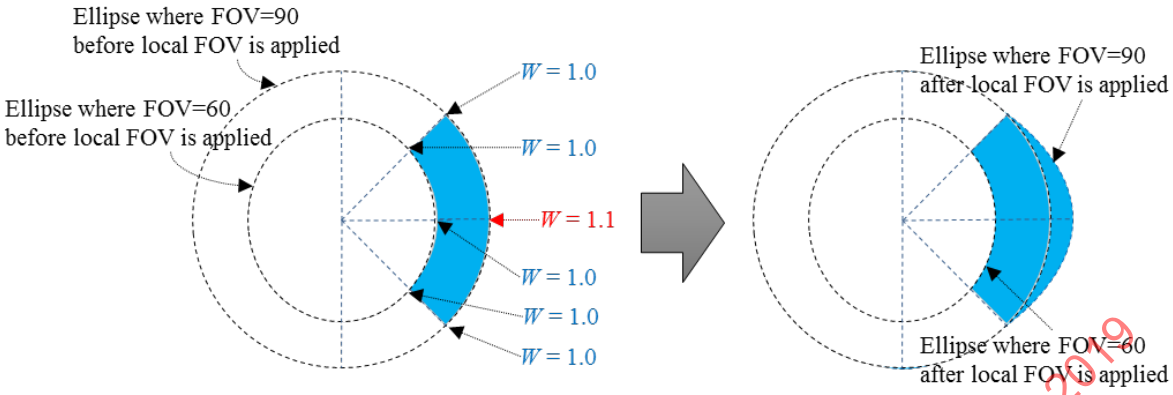


Figure 16 – Example of local FOV with a_s, a_e, and a_d equal to 45°, 135°, and 45°, repectively, and r_s, r_e, and r_d equal to 0.5, 1.0, and 0.5, respectively

`num_deadzones` is an integer that specifies the number of dead zones in each coded picture this structure applies to.

`deadzone_left_horizontal_offset`, `deadzone_top_vertical_offset`, `deadzone_width`, and `deadzone_height` are integer values that specify the position and size of the deadzone rectangular area in which the pixels are not usable. `deadzone_left_horizontal_offset` and `deadzone_top_vertical_offset` specify the horizontal and vertical coordinates, respectively, in luma samples, of the upper left corner of the deadzone in the coded picture. `deadzone_width` and `deadzone_height` specify the width and height, respectively, in luma samples, of the deadzone. To save bits for representing the video, all pixels within a deadzone should be set to the same pixel value, e.g., all black.

7 Omnidirectional media storage and metadata signalling in the ISOBMFF

7.1 Generic extensions to the ISOBMFF

7.1.1 Stereoscopic video track grouping

7.1.1.1 Definition

`TrackGroupTypeBox` with `track_group_type` equal to `'ster'` indicates that this track is either the left or right view of a stereo pair suitable for playback on a stereoscopic display.

The tracks that have the same value of `track_group_id` within `StereoVideoGroupBox` form a stereo pair, and there shall be no more than two of such tracks for the same value of `track_group_id`.

NOTE 1 Usually there are two tracks indicated to be a stereo pair with the `StereoVideoGroupBox` having the same value of `track_group_id`. However, only one track could be associated with a stereo pair in specific cases. For example, the file could be edited in a manner that one of the tracks forming a stereo pair gets removed. In another example, only one of the tracks of a stereo pair is selected for transmission, e.g. using DASH.

NOTE 2 When a track contains one or more `StereoVideoGroupBox`es within the `TrackGroupBox`, the respective Representation in a DASH MPD could use the multiple views scheme of DASH with `@value` including `r`***`ID`*** (when `left_view_flag` is equal to 0) or `l`***`ID`*** (when `left_view_flag` is equal to 1), where ***`ID`*** could be equal to the `track_group_id` provided that all content of the MPD originates from the same file or the uniqueness of `track_group_id` values across multiple files is ensured.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.1.1.2 Syntax

```
aligned(8) class StereoVideoGroupBox extends TrackGroupTypeBox('ster') {
   unsigned int(1) left_view_flag;
   bit(31) reserved = 0;
}
```

7.1.1.3 Semantics

`left_view_flag` equal to 0 indicates the right view of a stereo pair, and `left_view_flag` equal to 1 indicates the left view of a stereo pair. When there are two tracks with the same value of `track_group_id`, the value of `left_view_flag` shall differ.

7.1.2 Indication of `track_group_id` uniqueness

Bit 0 of the `flags` (with bit 0 being the least significant bit) of the `TrackGroupTypeBox` is used to indicate the uniqueness of `track_group_id`. The semantics of the flag is specified as follows:

(`flags` & 1) equal to 1 in a `TrackGroupTypeBox` of a particular `track_group_type` indicates that `track_group_id` in that `TrackGroupTypeBox` is not equal to any `track_ID` value and is not equal to `track_group_id` of any `TrackGroupTypeBox` with a different `track_group_type`. When (`flags` & 1) is equal to 1 in a `TrackGroupTypeBox` with particular values of `track_group_type` and `track_group_id`, (`flags` & 1) shall be equal to 1 in all `TrackGroupTypeBoxes` of the same values of `track_group_type` and `track_group_id`, respectively.

7.1.3 Updated semantics of `track_IDs` of the track reference box

The semantics of `track_IDs` of the `TrackReferenceBox` is changed from

`track_IDs` is an array of integers providing the track identifiers of the referenced tracks. The value 0 shall not be present. A given value shall not be duplicated in the array.

to

`track_IDs` is an array of integers providing the track identifiers of the referenced tracks or `track_group_id` values of the referenced track groups. Each value of `track_IDs[i]`, where i is a valid index to the `track_IDs[]` array, is an integer that provides a reference from the containing track to the track with `track_ID` equal to `track_IDs[i]` or to the track group with both `track_group_id` equal to `track_IDs[i]` and (`flags` & 1) of `TrackGroupTypeBox` equal to 1. When a `track_group_id` value is referenced, the track reference applies to each track of the referenced track group individually unless stated otherwise in the semantics of particular track reference types. The value 0 shall not be present. A given value shall not be duplicated in the array.

7.1.4 Indication of a track not intended to be presented alone

Bit 4 of the `flags` (with bit 0 being the least significant bit) of the `TrackHeaderBox` is used to indicate whether a track is not intended to be presented alone, e.g., due to that the track represents only a small portion of a videos scene. The semantics of the flag is specified as follows:

`track_not_intended_for_presentation_alone`: Indicates that the track is not intended to be presented alone without other tracks. Flag value is 0x000010. The flag not being set (i.e., `flags` & 0x000010 is equal to 0) indicates that the track may or may not be intended to be presented alone without other tracks.

7.1.5 Timed metadata tracks

7.1.5.1 Association with media tracks

When a timed metadata track is linked to one or more media tracks with a `'cdsc'` track reference, it describes each media track individually.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

7.1.6 Compatible scheme type box

7.1.6.1 Definition

Box Type: 'csch'
Container: RestrictedSchemeInfoBox
Mandatory: No
Quantity: Zero or more

CompatibleSchemeTypeBox identifies a scheme type that the track conforms to.

```
aligned(8) class CompatibleSchemeTypeBox extends FullBox('csch', 0, flags) {
   // identical syntax to SchemeTypeBox
   unsigned int(32)  scheme_type;          // 4CC identifying the scheme
   unsigned int(32)  scheme_version;       // scheme version
   if (flags & 0x000001) {
      unsigned int(8)   scheme_uri[];            // browser uri
   }
}
```

7.1.6.2 Semantics

The semantics of the syntax elements are identical to the semantics of the syntax elements with the same name in SchemeTypeBox.

7.1.7 Multiple transformations for a single transformed media track

A transformed media track may have undergone several transformations of different types and shall not have undergone more than one transformation of any particular type.

EXAMPLE A transformed track could be both protected and restricted.

The following process applies to conclude the untransformed sample entry type of a transformed media track:

1. Let schemeInfoContainerBox be ProtectionSchemeInfoBox, RestrictedSchemeInfoBox, or CompleteTrackInfoBox when the track sample entry type indicates an encrypted, restricted, or incomplete media track, respectively.
2. Let dataFormat be equal to the data_format value of OriginalFormatBox of schemeInfoContainerBox.
3. If dataFormat indicates a transformed media track, schemeInfoContainerBox is updated to be ProtectionSchemeInfoBox, RestrictedSchemeInfoBox, or CompleteTrackInfoBox when dataFormat indicates an encrypted, restricted, or incomplete media track, respectively. The process continues from step 2.
4. Otherwise (dataFormat does not indicate a transformed media track), the untransformed sample entry type is concluded to be equal to dataFormat.

7.1.8 The 'codecs' parameter for a transformed media track

For a transformed media track, the 'codecs' parameter is a comma-separated list of one or more list items. Each list item is formed from a sample entry of the transformed media track by concatenating the track sample entry type (e.g., 'resv'), a scheme type string, zero or more pairs of sample entry types and scheme type strings of transformed media, and an original format string (e.g., 'avc1'), each separated by '.'.

The following process applies to derive the value for a list item of the codecs parameter of a transformed media track:

1. The value of the codecs parameter is initialized to be an empty string.
2. Let schemeInfoContainerBox be ProtectionSchemeInfoBox, RestrictedSchemeInfoBox, or CompleteTrackInfoBox when the track sample entry type indicates an encrypted, restricted, or incomplete media track, respectively.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

3. Let dataFormat be equal to the `data_format` value of `OriginalFormatBox` of schemeInfoContainerBox. The value of the `codecs` parameter is appended by dataFormat followed by a dot ('.') character.
4. When dataFormat indicates a transformed media track, the value of the `codecs` parameter is appended by the `scheme_type` four-character code contained in the `SchemeTypeBox` of schemeInfoContainerBox.
5. When schemeInfoContainerBox contains instances of `CompatibleSchemeTypeBox`, the following applies for each instance of the `CompatibleSchemeTypeBox` of schemeInfoContainerBox in any order: the value of the `codecs` parameter is appended by a plus character ('+') followed by the `scheme_type` four-character code contained in the `CompatibleSchemeTypeBox`.
6. The value of the `codecs` parameter is appended by a dot ('.') character.
7. If dataFormat indicates a transformed media track, schemeInfoContainerBox is updated to be `ProtectionSchemeInfoBox`, `RestrictedSchemeInfoBox`, or `CompleteTrackInfoBox` when dataFormat indicates an encrypted, restricted, or incomplete media track, respectively. The process continues from step 3.
8. Otherwise (dataFormat does not indicate a transformed media track), the value of the `codecs` parameter is appended by the original format string.

The original format string starts with the untransformed sample entry type indicating a coding format and follows the specifications of the `'codecs'` string for that coding format.

In the process above, steps 4 and 5 describe the derivation of a scheme type string.

For example, the value of the `'codecs'` parameter might be `'resv.podv.erpv.avc1'`.

7.1.9 Track type box

7.1.9.1 Definition

Box Type: `'ttyp'`
Container: `TrackBox`
Mandatory: No
Quantity: Zero or one

The payload of `TrackTypeBox` has the same syntax as the payload of `FileTypeBox`. The content of an instance of `TrackTypeBox` shall be such that it would apply as the content of `FileTypeBox`, if all other tracks of the file were removed and only the track containing this box remained in the file.

NOTE `TrackTypeBox` could be used in specifying media profiles or track-specific brands.

7.1.9.2 Syntax

```
aligned(8) class TrackTypeBox extends FullBox('ttyp', 0, 0) {
      unsigned int(32)  major_brand;
      unsigned int(32)  minor_version;
      unsigned int(32)  compatible_brands[];    // to end of the box
}
```

NOTE The payload syntax of `TrackTypeBox` is identical to that of `FileTypeBox`.

7.1.9.3 Semantics

The semantics of the syntax elements are identical to the semantics of the syntax elements with the same name in `FileTypeBox`, if all other tracks of the file were removed and only the track containing this box remained in the file.

7.1.10 Clarifications on the stereo video box

This document uses `StereoVideoBox` as specified in ISO/IEC 14496-12 with the following additional specifications:

– The definition of `StereoVideoBox` is changed from

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

Box Type: 'stvi'
Container: SchemeInformationBox
Mandatory: Yes (when the SchemeType is 'stvi')
Quantity: One

to

Box Type: 'stvi'
Container: SchemeInformationBox
Mandatory: Yes (when the SchemeType is 'stvi'), no (otherwise)
Quantity: Zero or one

- When stereo_scheme is equal to 4, **PackedContentInterpretationType** specified in ISO/IEC 23091-2 is inferred to be equal to 1.
- When the value of stereo_indication_type indicates the temporal interleaving frame packing arrangement and the display system in use presents two views simultaneously, readers should implicitly set the composition time for constituent picture 0 to coincide with the composition time for constituent picture 1.
- When the value of stereo_indication_type indicates the temporal interleaving frame packing arrangement, each sync sample and each SAP sample with SAP_type in the range of 1 to 3, inclusive, indicated by the stream access point sample group represent constituent picture 0 followed, in composition time order, by samples representing constituent pictures 1 and 0 in an alternating manner up to but excluding the next sync sample or SAP sample with SAP_type in the range of 1 to 3, inclusive, indicated by the stream access point sample group.

7.2 Generic extensions to ISO/IEC 14496-15

7.2.1 Alternative extraction source track grouping

Members of the track group with track_group_type equal to 'alte' are alternatives to be used as a source for 'scal' or 'sabt' track reference. The value of (flags & 1) shall be equal to 1 in a TrackGroupTypeBox of type 'alte' to indicate the uniqueness of track_group_id as specified in subclause 7.1.2.

A 'scal' or 'sabt' track reference may refer to a track_group_id value of an 'alte' track group. As implied by the general semantics of a track reference to a track_group_id as specified in subclause 7.1.3, any single track of an 'alte' track group is a valid source for extraction as specified in ISO/IEC 14496-15:2017, Clauses A.3 and A.7 or for bitstream reconstruction from tile tracks as specified in ISO/IEC 14496-15:2017, subclause 10.5.4.

7.2.2 Tile base track association with coverage information box and timed metadata data track

When a timed metadata track is linked to a tile base track with a 'cdsc' track reference, it describes the HEVC video bitstream carried by the tile base track and all the associated tile tracks.

7.3 OMAF-specific extensions to the ISOBMFF

7.3.1 Sync samples in timed metadata tracks

For timed metadata tracks specified in this document, a sample in a timed metadata track is defined as a sync sample if and only if at least one of media samples in the referenced media tracks having the same decoding time is a sync sample.

7.4 OMAF-specific extensions to ISO/IEC 14496-15

7.4.1 Coverage information box in a tile base track

When CoverageInformationBox is included in ProjectedOmniVideoBox of a tile base track, it provides information on the area on the sphere covered by the content that is represented by all the tile tracks associated with the tile base track.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.5 Structures and semantics that are common for video tracks and image items

7.5.1 Semantics of sample locations within a decoded picture

7.5.1.1 Relation of decoded pictures to global coordinate axes (informative)

Figure 17 illustrates the conversions from a spherical picture to a packed picture that could be used in content authoring and the corresponding conversions from a packed picture to a spherical picture to be rendered that could be used in an OMAF player. The example in this clause is described for a packed picture that appears in a projected omnidirectional video track. Similar description could be derived for an image item.

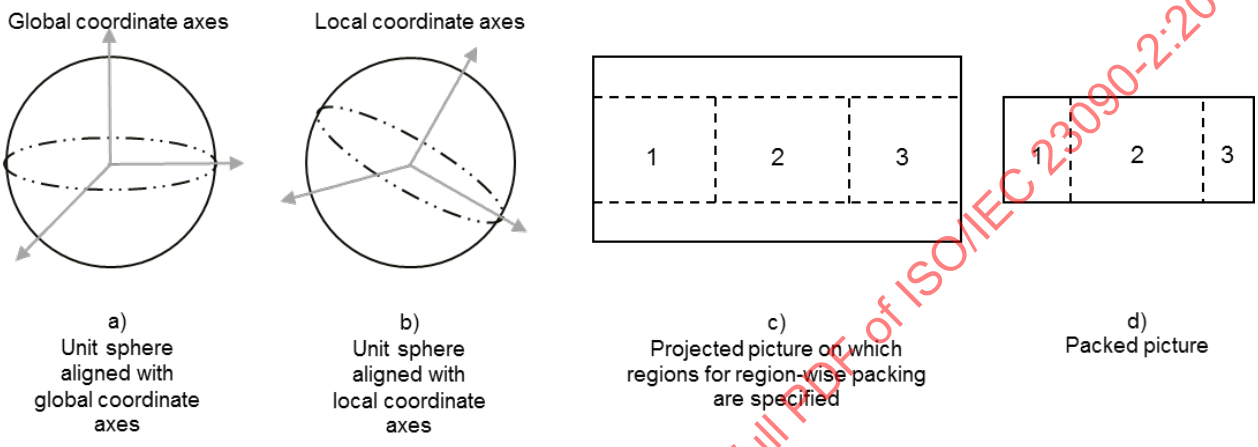


Figure 17 – Example of processing stages to derive a packed picture from a spherical image or vice versa

The content authoring could include the following ordered steps:

- The source images provided as input are stitched to generate a sphere picture on the unit sphere per the global coordinate axes as indicated in Figure 17a.
- The unit sphere is then rotated relative to the global coordinate axes, as indicated in Figure 17b. The amount of rotation to convert from the local coordinate axes to the global coordinate axes is specified by the rotation angles indicated in the `RotationBox`. The local coordinate axes of the unit sphere are the axes of the coordinate system that has been rotated. The absence of `RotationBox` indicates that the local coordinate axes are the same as the global coordinate axes.
- As illustrated in Figure 17c, the spherical picture on the rotated unit sphere is then converted to a two-dimensional projected picture, for example using the equirectangular projection specified in subclause 5.2.1. When spatial packing of stereoscopic content is applied, two spherical pictures for the two views are converted to two constituent pictures, after which frame packing is applied to pack the two constituent pictures to one projected picture.
- Rectangular region-wise packing could be applied to obtain a packed picture from the projected picture. One example of packing is depicted in Figure 17c and Figure 17d. The dashed rectangles in Figure 17c indicate the projected regions on a projected picture, and the respective areas in Figure 17d indicate the corresponding packed regions. In this example, projected regions 1 and 3 are horizontally downsampled, while projected region 2 is kept at its original resolution.

`CoverageInformationBox` could be used to indicate which part of the sphere is covered by the packed picture.

In order to map sample locations of a packed picture (such as that in Figure 17d) to a unit sphere used in rendering (Figure 17a), the OMAF player could perform the following ordered steps:

- A packed picture, such as that in Figure 17d, is obtained as a result of decoding a picture from a video track or an image item.
- If needed, chroma sample arrays of the packed picture are upsampled to the resolution of the luma sample array of the packed picture, and colour space conversion could also be performed.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

– If region-wise packing is indicated, the sample locations of the packed picture are converted to sample locations of the respective projected picture, such as that in Figure 17c, as specified in subclause 5.4.2. Otherwise, the projected picture is identical to the packed picture.

– If spatial frame packing of the projected picture is indicated, the sample locations of the projected picture are converted to sample locations of the respective constituent picture of the projected picture, as specified in subclause 7.5.1.3. Otherwise, the constituent picture of the projected picture is identical to the projected picture.

– The sample locations of a constituent picture the projected picture are converted to sphere coordinates that are relative to local coordinate axes, as specified for the omnidirectional projection format being used in subclause 5.2. The resulting sample locations correspond to a sphere picture depicted in Figure 17b.

– If rotation is indicated, the sphere coordinates relative to the local coordinate axes are converted to sphere coordinates relative to the global coordinate axes as specified in subclause 5.3. Otherwise, the global coordinate axes are identical to the local coordinate axes.

The overall process for mapping of luma sample locations within a decoded picture to sphere coordinates relative to the global coordinate axes is normatively specified in subclause 7.5.1.2.

7.5.1.2 Mapping of luma sample locations within a decoded picture to sphere coordinates relative to the global coordinate axes

This clause specifies the semantics of luma sample locations within a decoded picture to sphere coordinates relative to the global coordinate axes. The decoded picture may be of any of the following:

- For video, the decoded picture is the decoding output resulting from a sample of the video track.
- For an image item, the decoded picture is a reconstructed image of the image item.

offsetX is set equal to 0.5 and offsetY is set equal to 0.5.

If RegionWisePackingFlag is equal to 1, the following applies for each packed region n in the range of 0 to NumRegions − 1, inclusive:

– For each sample location (xPackedPicture, yPackedPicture) belonging to the n-th packed region with PackingType[n] equal to 0 (i.e., with rectangular region-wise packing), the following applies:

  – The corresponding sample location (xProjPicture, yProjPicture) of the projected picture is derived as follows:

    – x is set equal to xPackedPicture – PackedRegLeft[n].

    – y is set equal to yPackedPicture – PackedRegTop[n].

    – Subclause 5.4.2 is invoked with x, y, PackedRegWidth[n], PackedRegHeight[n], ProjRegWidth[n], ProjRegHeight[n], TransformType[n], offsetX, and offsetY as inputs, and the output is assigned to sample location (hPos, vPos).

    – xProjPicture is set equal to ProjRegLeft[n] + hPos.

    – When SideBySideFlag is equal to 0, and when xProjPicture is greater than or equal to `proj_picture_width`, xProjPicture is set equal to xProjPicture − `proj_picture_width`.

    – When SideBySideFlag is equal to 1, the following applies:

      – When ProjRegLeft[n] is less than `proj_picture_width` / 2 and xProjPicture is greater than or equal to `proj_picture_width` / 2, xProjPicture is set equal to xProjPicture − `proj_picture_width` / 2.

      – When ProjRegLeft[n] is greater than or equal to `proj_picture_width` / 2 and xProjPicture is greater than or equal to `proj_picture_width`, xProjPicture is set equal to xProjPicture − `proj_picture_width` / 2.

    – yProjPicture is set equal to ProjRegTop[n] + vPos.

  – Subclause 7.5.1.3 is invoked with xProjPicture, yProjPicture, ConstituentPicWidth, and ConstituentPicHeight as inputs, and the outputs indicating the sphere coordinates and the constituent frame index (for frame-packed

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

stereoscopic video) for the luma sample location (xPackedPicture, yPackedPicture) belonging to the n-th packed region in the decoded picture.

Otherwise (RegionWisePackingFlag is equal to 0), the following applies for each sample location (x, y) within the decoded picture:

– xProjPicture is set equal to x + offsetX.

– yProjPicture is set equal to y + offsetY.

– Subclause 7.5.1.3 is invoked with xProjPicture, yProjPicture, ConstituentPicWidth, and ConstituentPicHeight as inputs, and the outputs indicating the sphere coordinates and the constituent frame index (for frame-packed stereoscopic video) for the sample location (x, y) within the decoded picture.

7.5.1.3 Conversion from a sample location in a projected picture to sphere coordinates relative to the global coordinate axes

Inputs to this process are

– the centre point of a sample location (xProjPicture, yProjPicture) within a projected picture, where xProjPicture and yProjPicture are in relative projected picture sample units and may have non-integer real values, and

– pictureWidth and pictureHeight, which are the width and height, respectively, of a monoscopic projected luma picture, in relative projected picture sample units.

NOTE 1 The projected picture for which the sample location (xProjPicture, yProjPicture) is given as input could be a spatially frame-packed picture.

Outputs of this process are:

– sphere coordinates (azimuthGlobal, elevationGlobal), in units of degrees relative to the global coordinate axes, and

– when SpatiallyPackedStereoFlag is equal to 1, the index of the constituent picture (constituentPicture) equal to 0 or 1.

NOTE 2 When the temporal interleaving packing arrangement is in use, the projected picture is associated with the left view or right view as specified in subclause 7.1.10.

The outputs are derived with the following ordered steps:

– If xProjPicture is greater than or equal to pictureWidth or yProjPicture is greater than or equal to pictureHeight, the following applies:

  – constituentPicture is set equal to 1.

  – If xProjPicture is greater than or equal to pictureWidth, xProjPicture is set to xProjPicture − pictureWidth.

  – If yProjPicture is greater than or equal to pictureHeight, yProjPicture is set to yProjPicture − pictureHeight.

– Otherwise, constituentPicture is set equal to 0.

– Depending on the projection format, the following applies:

  – When the projection format is the equirectangular projection, subclause 5.2.2 is invoked with pictureWidth, pictureHeight, xProjPicture, and yProjPicture as inputs, and the output is assigned to azimuthLocal, elevationLocal.

  – When the projection format is the cubemap projection, subclause 5.2.3 is invoked with pictureWidth, pictureHeight, xProjPicture, and yProjPicture as inputs, and the output is assigned to azimuthLocal, elevantionLocal.

– If RotationFlag is equal to 1, subclause 5.3 is invoked with azimuthLocal, elevantionLocal, `rotation_yaw` ÷ 2^{16}, `rotation_pitch` ÷ 2^{16}, and `rotation_roll` ÷ 2^{16} as inputs, and the output is assigned to azimuthGlobal and elevationGlobal.

– Otherwise, azimuthGlobal is set equal to azimuthLocal and elevationGlobal is set equal to elevationLocal.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

7.5.2 Projection format structure

7.5.2.1 Syntax

```
aligned(8) class ProjectionFormatStruct() {
   bit(3) reserved = 0;
   unsigned int(5) projection_type;
}
```

7.5.2.2 Semantics

`projection_type` indicates the type of the mapping of the projected picture onto the spherical coordinate system as specified in subclause 5.1. The values of `projection_type` and their semantics are specified in Table 8.

7.5.3 Region-wise packing structure

7.5.3.1 Definition

`RegionWisePackingStruct` specifies the mapping between packed regions and the respective projected regions and specifies the location and size of the guard bands, if any.

NOTE Among other information the `RegionWisePackingStruct` also provides the content coverage information in the 2D Cartesian picture domain.

A decoded picture in the semantics of this clause is either one of the following depending on the container for this syntax structure:

- For video, the decoded picture is the decoding output resulting from a sample of the video track.
- For an image item, the decoded picture is a reconstructed image of the image item.

The content of `RegionWisePackingStruct` is informatively summarized below, while the normative semantics follow subsequently in this clause:

- The width and height of the projected picture are explicitly signalled with `proj_picture_width` and `proj_picture_height`, respectively.
- The width and height of the packed picture are explicitly signalled with `packed_picture_width` and `packed_picture_height`, respectively.
- When the projected picture is stereoscopic and has the top-bottom or side-by-side frame packing arrangement, `constituent_picture_matching_flag` equal to 1 specifies that
  - the projected region information, packed region information, and guard band region information in this syntax structure apply individually to each constituent picture,
  - the packed picture and the projected picture have the same stereoscopic frame packing format, and
  - the number of projected regions and packed regions is double of that indicated by the value of `num_regions` in the syntax structure.
- `RegionWisePackingStruct` contains a loop, in which a loop entry corresponds to the respective projected regions and packed regions in both constituent pictures (when `constituent_picture_matching_flag` equal to 1) or to a projected region and the respective packed region (when `constituent_picture_matching_flag` equal to 0), and the loop entry the contains the following:
  - a flag indicating the presence of guard bands for the packed region,
  - the packing type (however, only rectangular region-wise packing is specified in this document),
  - the mapping between a projected region and the respective packed region in the rectangular region packing structure `RectRegionPacking(i)`,
  - when guard bands are present, the guard band structure for the packed region `GuardBand(i)`.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

The content of the rectangular region packing structure `RectRegionPacking(i)` is informatively summarized below, while the normative semantics follow subsequently in this clause:

- `proj_reg_width[i]`, `proj_reg_height[i]`, `proj_reg_top[i]`, and `proj_reg_left[i]` specify the width, height, top offset, and left offset, respectively, of the i-th projected region.
- `transform_type[i]` specifies the rotation and mirroring, if any, that are applied to the i-th packed region to remap it to the i-th projected region.
- `packed_reg_width[i]`, `packed_reg_height[i]`, `packed_reg_top[i]`, and `packed_reg_left[i]` specify the width, height, the top offset, and the left offset, respectively, of the i-th packed region.

The content of the guard band structure `GuardBand(i)` is informatively summarized below, while the normative semantics follow subsequently in subclause 7.5.3:

- `left_gb_width[i]`, `right_gb_width[i]`, `top_gb_height[i]`, or `bottom_gb_height[i]` specify the guard band size on the left side of, the right side of, above, or below, respectively, the i-th packed region.
- `gb_not_used_for_pred_flag[i]` indicates if the encoding was constrained in a manner that guards bands are not used as a reference in the inter prediction process.
- `gb_type[i][j]` specifies the type of the guard bands for the i-th packed region.

Figure 18 illustrates an example of the position and size of a projected region within a projected picture (on the left side) as well as that of a packed region within a packed picture with guard bands (on the right side). This example applies when the value of `constituent_picture_matching_flag` is equal to 0.

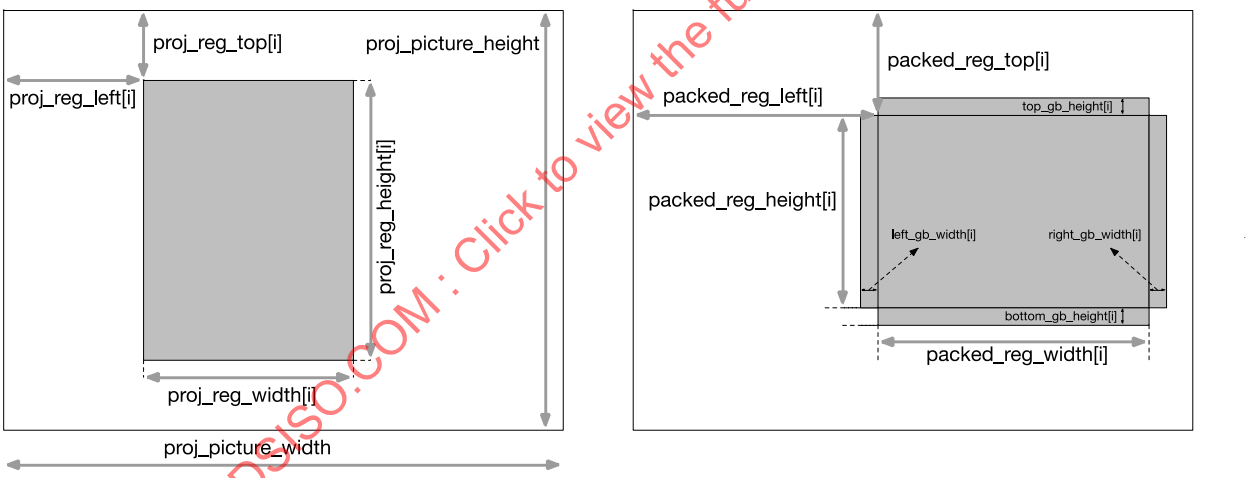


Figure 18 – Projected region and the corresponding packed region with guard bands (informative)

Subclause 7.5.3 is organized as follows:

- The syntax and semantics of the rectangular region packing structure are specified in subclauses 7.5.3.2 and 7.5.3.3, respectively.
- The syntax and semantics of the guard band structure are specified in subclauses 7.5.3.4 and 7.5.3.5, respectively.
- The syntax and semantics of the region-wise packing structure are specified in subclauses 7.5.3.6 and 7.5.3.7, respectively.
- Subclause 7.5.3.8 derives variables from syntax element values of the rectangular region packing, guard band, region-wise packing structures. Subclause 7.5.3.8 also uses the variables to specify constraints for the syntax element values. The variables are also used in other clauses.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.5.3.2 Syntax of the rectangular region packing structure

```
aligned(8) class RectRegionPacking(i) {
   unsigned int(32) proj_reg_width[i];
   unsigned int(32) proj_reg_height[i];
   unsigned int(32) proj_reg_top[i];
   unsigned int(32) proj_reg_left[i];
   unsigned int(3)  transform_type[i];
   bit(5) reserved = 0;
   unsigned int(16) packed_reg_width[i];
   unsigned int(16) packed_reg_height[i];
   unsigned int(16) packed_reg_top[i];
   unsigned int(16) packed_reg_left[i];
}
```

7.5.3.3 Semantics of the rectangular region packing structure

`proj_reg_width[i]`, `proj_reg_height[i]`, `proj_reg_top[i]`, and `proj_reg_left[i]` specify the width, height, top offset, and left offset, respectively, of the i-th projected region, either within the projected picture (when `constituent_picture_matching_flag` is equal to 0) or within the constituent picture of the projected picture (when `constituent_picture_matching_flag` is equal to 1). `proj_reg_width[i]`, `proj_reg_height[i]`, `proj_reg_top[i]` and `proj_reg_left[i]` are indicated in relative projected picture sample units.

NOTE 1 Two projected regions could partially or entirely overlap with each other. When there is an indication of quality difference, e.g., by a region-wise quality ranking indication, then for the overlapping area of any two overlapping projected regions, the packed region corresponding to the projected region that is indicated to have higher quality is expected to be used for rendering.

`transform_type[i]` specifies the rotation and mirroring that is applied to the i-th packed region to remap it to the i-th projected region. When `transform_type[i]` specifies both rotation and mirroring, rotation is applied before mirroring for converting sample locations of a packed region to sample locations of a projected region. The following values are specified:

0: no transform

1: mirroring horizontally

2: rotation by 180 degrees (counter-clockwise)

3: rotation by 180 degrees (counter-clockwise) before mirroring horizontally

4: rotation by 90 degrees (counter-clockwise) before mirroring horizontally

5: rotation by 90 degrees (counter-clockwise)

6: rotation by 270 degrees (counter-clockwise) before mirroring horizontally

7: rotation by 270 degrees (counter-clockwise)

NOTE 2 Subclause 5.4.2 specifies the semantics of `transform_type[i]` for converting a sample location of a packed region in a packed picture to a sample location of a projected region in a projected picture.

`packed_reg_width[i]`, `packed_reg_height[i]`, `packed_reg_top[i]`, and `packed_reg_left[i]` specify the width, height, the offset, and the left offset, respectively, of the i-th packed region, either within the packed picture (when `constituent_picture_matching_flag` is equal to 0) or within each constituent picture of the packed picture (when `constituent_picture_matching_flag` is equal to 1). `packed_reg_width[i]`, `packed_reg_height[i]`, `packed_reg_top[i]`, and `packed_reg_left[i]` are indicated in relative packed picture sample units. `packed_reg_width[i]`, `packed_reg_height[i]`, `packed_reg_top[i]`, and `packed_reg_left[i]` shall represent integer horizontal and vertical coordinates of luma sample units within the decoded pictures.

NOTE 3 Two packed regions could partially or entirely overlap with each other.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.5.3.4 Syntax of the guard band structure

```
aligned(8) class GuardBand(i) {
   unsigned int(8) left_gb_width[i];
   unsigned int(8) right_gb_width[i];
   unsigned int(8) top_gb_height[i];
   unsigned int(8) bottom_gb_height[i];
   unsigned int(1) gb_not_used_for_pred_flag[i];
   for (j = 0; j < 4; j++)
      unsigned int(3) gb_type[i][j];
   bit(3) reserved = 0;
}
```

7.5.3.5 Semantics of the guard band structure

`left_gb_width[i]` specifies the width of the guard band on the left side of the i-th packed region in relative packed picture sample units. When the decoded picture has 4:2:0 or 4:2:2 chroma format, `left_gb_width[i]` shall correspond to an even number of luma samples within the decoded picture.

`right_gb_width[i]` specifies the width of the guard band on the right side of the i-th packed region in relative packed picture sample units. When the decoded picture has 4:2:0 or 4:2:2 chroma format, `right_gb_width[i]` shall correspond to an even number of luma samples within the decoded picture.

`top_gb_height[i]` specifies the height of the guard band above the i-th packed region in relative packed picture sample units. When the decoded picture has 4:2:0 chroma format, `top_gb_height[i]` shall correspond to an even number of luma samples within the decoded picture.

`bottom_gb_height[i]` specifies the height of the guard band below the i-th packed region in relative packed picture sample units. When the decoded picture has 4:2:0 chroma format, `bottom_gb_height[i]` shall correspond to an even number of luma samples within the decoded picture.

When `GuardBand(i)` is present, at least one of `left_gb_width[i]`, `right_gb_width[i]`, `top_gb_height[i]`, or `bottom_gb_height[i]` shall be greater than 0.

`gb_not_used_for_pred_flag[i]` equal to 0 specifies that the guard bands may or may not be used in the inter prediction process. `gb_not_used_for_pred_flag[i]` equal to 1 specifies that the sample values of the guard bands are not used in the inter prediction process.

NOTE 1 When `gb_not_used_for_pred_flag[i]` is equal to 1, the sample values within guard bands in decoded pictures could be rewritten even if the decoded pictures were used as references for inter prediction of subsequent pictures to be decoded. For example, the content of a packed region could be seamlessly expanded to its guard band with decoded and re-projected samples of another packed region.

`gb_type[i][j]` specifies the type of the guard bands for the i-th packed region as follows, with j equal to 0, 1, 2, or 3 indicating that the semantics below apply to the left, right, top, or bottom edge, respectively, of the packed region:

- `gb_type[i][j]` equal to 0 specifies that the content of the guard bands in relation to the content of the packed regions is unspecified. When `gb_not_used_for_pred_flag[i]` is equal to 0, `gb_type[i][j]` shall not be equal to 0.
- `gb_type[i][j]` equal to 1 specifies that the content of the guard bands suffices for interpolation of sub-pixel values within the packed region and less than one pixel outside of the boundary of the packed region.

NOTE 2 `gb_type[i][j]` equal to 1 could be used when the boundary samples of a packed region have been copied horizontally or vertically to the guard band.

- `gb_type[i][j]` equal to 2 specifies that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the packed region and is on the surface of the packed region at quality that gradually changes from the picture quality of the packed region to that of the spherically adjacent packed region.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

- `gb_type[i][j]` equal to 3 specifies that the content of the guard bands represents actual picture content that is spherically adjacent to the content in the packed region and is on the surface of the packed region at the picture quality of the packed region.

- `gb_type[i][j]` values greater than 3 are reserved.

7.5.3.6 Syntax the region-wise packing structure

```
aligned(8) class RegionWisePackingStruct() {
   unsigned int(1) constituent_picture_matching_flag;
   bit(7) reserved = 0;
   unsigned int(8) num_regions;
   unsigned int(32) proj_picture_width;
   unsigned int(32) proj_picture_height;
   unsigned int(16) packed_picture_width;
   unsigned int(16) packed_picture_height;
   for (i = 0; i < num_regions; i++) {
      bit(3) reserved = 0;
      unsigned int(1) guard_band_flag[i];
      unsigned int(4) packing_type[i];
      if (packing_type[i] == 0) {
         RectRegionPacking(i);
         if (guard_band_flag[i])
            GuardBand(i);
      }
   }
}
```

7.5.3.7 Semantics of the region-wise packing structure

`constituent_picture_matching_flag` equal to 1 specifies that the projected region information, packed region information, and guard band region information in this syntax structure apply individually to each constituent picture and that the packed picture and the projected picture have the same stereoscopic frame packing format. `constituent_picture_matching_flag` equal to 0 specifies that the projected region information, packed region information, and guard band region information in this syntax structure apply to the projected picture. When SpatiallyPackedStereoFlag is equal to 0, `constituent_picture_matching_flag` shall be equal to 0.

NOTE 1 For the stereoscopic content that uses equivalent region-wise packing for the constituent pictures, setting this flag equal to 1 allows more compact signalling of region-wise packing information.

`num_regions` specifies the number of packed regions when `constituent_picture_matching_flag` is equal to 0. Value 0 is reserved. When `constituent_picture_matching_flag` is equal to 1, the total number of packed regions is equal to 2 * `num_regions` and the information in `RectRegionPacking(i)` and `GuardBand(i)` applies to each constituent picture of the projected picture and the packed picture.

`proj_picture_width` and `proj_picture_height` specify the width and height, respectively, of the projected picture, in relative projected picture sample units. `proj_picture_width` and `proj_picture_height` shall both be greater than 0.

NOTE 2 The same sampling grid, width and height are used for the luma sample array and the chroma sample arrays of the projected picture.

`packed_picture_width` and `packed_picture_height` specify the width and height, respectively, of the packed picture, in relative packed picture sample units. `packed_picture_width` and `packed_picture_height` shall both be greater than 0.

`guard_band_flag[i]` equal to 0 specifies that the i-th packed region has no guard bands. `guard_band_flag[i]` equal to 1 specifies that the i-th packed region has at least one guard band.

`packing_type[i]` specifies the type of region-wise packing. The values of `packing_type[i]` and their semantics are specified in Table 9.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`RectRegionPacking(i)` specifies the region-wise packing between the i-th packed region and the i-th projected region. The syntax and semantics of `RectRegionPacking(i)` are specified in subclauses 7.5.3.2 and 7.5.3.3, respectively.

`GuardBand(i)` specifies the guard bands for the i-th packed region. The syntax and semantics of `GuardBand(i)` are specified in subclauses 7.5.3.4 and 7.5.3.5, respectively.

7.5.3.8 Derivation of region-wise packing variables and constraints for the syntax elements of the region-wise packing structure

When the i-th packed region as specified by this `RegionWisePackingStruct` overlaps with the j-th packed region specified by the same `RegionWisePackingStruct`, the i-th and j-th projected regions shall reside in different constituent pictures for any values of i and j that are not equal to each other. The i-th packed region as specified by this `RegionWisePackingStruct` shall not overlap with any guard band specified by the same `RegionWisePackingStruct`.

The guard bands associated with the i-th packed region, if any, as specified by this `RegionWisePackingStruct` shall not overlap with any packed region specified by the same `RegionWisePackingStruct` or any other guard bands specified by the same `RegionWisePackingStruct`.

NOTE Projected regions are allowed to overlap. When projected regions overlap and a quality difference is indicated between the projected regions, e.g., by a region-wise quality ranking indication, the packed region that is indicated to have the highest quality among the packed regions corresponding to the projected regions that overlap should be used for rendering the overlapping area.

The variables NumRegions, PackedRegLeft[n], PackedRegTop[n], PackedRegWidth[n], PackedRegHeight[n], ProjRegLeft[n], ProjRegTop[n], ProjRegWidth[n], ProjRegHeight[n], TrasnformType[n], PackingType[n] are derived as follows:

- For n in the range of 0 to `num_regions` − 1, inclusive, the following applies:
  - PackedRegLeft[n] is set equal to `packed_reg_left[n]`.
  - PackedRegTop[n] is set equal to `packed_reg_top[n]`.
  - PackedRegWidth[n] is set equal to `packed_reg_width[n]`.
  - PackedRegHeight[n] is set equal to `packed_reg_height[n]`.
  - ProjRegLeft[n] is set equal to `proj_reg_left[n]`.
  - ProjRegTop[n] is set equal to `proj_reg_top[n]`.
  - ProjRegWidth[n] is set equal to `proj_reg_width[n]`.
  - ProjRegHeight[n] is set equal to `proj_reg_height[n]`.
  - TransformType[n] is set equal to `transform_type[n]`.
  - PackingType[n] is set equal to `packing_type[n]`.
- If `constituent_picture_matching_flag` is equal to 0, the following applies:
  - NumRegions is set equal to `num_regions`.
- Otherwise (`constituent_picture_matching_flag` is equal to 1), the following applies:
  - NumRegions is set equal to 2 * `num_regions`.
  - When TopBottomFlag is equal to 1, the following applies:
    - projLeftOffset and packedLeftOffset are both set equal to 0.
    - projTopOffset is set equal to `proj_picture_height` / 2 and packedTopOffset is set equal to `packed_picture_height` / 2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

- When SideBySideFlag is equal to 1, the following applies:
  - projLeftOffset is set equal to `proj_picture_width` / 2 and packedLeftOffset is set equal to `packed_picture_width` / 2.
  - projTopOffset and packedTopOffset are both set equal to 0.
- For n in the range of NumRegions / 2 to NumRegions − 1, inclusive, the following applies:
  - nIdx is set equal to n – NumRegions / 2.
  - PackedRegLeft[n] is set equal to `packed_reg_left[nIdx]` + packedLeftOffset.
  - PackedRegTop[n] is set equal to `packed_reg_top[nIdx]` + packedTopOffset.
  - PackedRegWidth[n] is set equal to `packed_reg_width[nIdx]`.
  - PackedRegHeight[n] is set equal to `packed_reg_height[nIdx]`.
  - ProjRegLeft[n] is set equal to `proj_reg_left[nIdx]` + projLeftOffset.
  - ProjRegTop[n] is set equal to `proj_reg_top[nIdx]` + projTopOffset.
  - ProjRegWidth[n] is set equal to `proj_reg_width[nIdx]`.
  - ProjRegHeight[n] is set equal to `proj_reg_height[nIdx]`.
  - TransformType[n] is set equal to `transform_type[nIdx]`.
  - PackingType[n] is set equal to `packing_type[nIdx]`.

For each value of n in the range of 0 to NumRegions − 1, inclusive, the values of ProjRegWidth[n], ProjRegHeight[n], ProjRegTop[n], and ProjRegLeft[n] are constrained as follows:

- ProjRegWidth[n] shall be in the range of 1 to `proj_picture_width`, inclusive.
- ProjRegHeight[n] shall be in the range of 1 to `proj_picture_height`, inclusive.
- ProjRegLeft[n] shall be in the range of 0 to `proj_picture_width` − 1, inclusive.
- ProjRegTop[n] shall be in the range of 0 to `proj_picture_height` − 1, inclusive.
- If ProjRegTop[n] is less than `proj_picture_height` / VerDiv1, the sum of ProjRegTop[n] and ProjRegHeight[n] shall be less than or equal to `proj_picture_height` / VerDiv1. Otherwise, the sum of ProjRegTop[n] and ProjRegHeight[n] shall be less than or equal to `proj_picture_height` / VerDiv1 * 2.

For each value of n in the range of 0 to NumRegions − 1, inclusive, the values of PackedRegWidth[n], PackedRegHeight[n], PackedRegTop[n], and PackedRegLeft[n] are constrained as follows:

- PackedRegWidth[n] shall be in the range of 1 to `packed_picture_width`, inclusive.
- PackedRegHeight[n] shall be in the range of 1 to `packed_picture_height`, inclusive.
- PackedRegLeft[n] shall be in the range of 0 to `packed_picture_width` − 1, inclusive.
- PackedRegTop[n] shall be in the range of 0 to `packed_picture_height` − 1, inclusive.
- If PackedRegLeft[n] is less than `packed_picture_width` / HorDiv1, the sum of PackedRegLeft[n] and PackedRegWidth[n] shall be less than or equal to `packed_picture_width` / HorDiv1. Otherwise, the sum of PackedRegLeft[n] and PackedRegWidth[n] shall be less than or equal to `packed_picture_width` / HorDiv1 * 2.
- If PackedRegTop[n] is less than `packed_picture_height` / VerDiv1, the sum of PackedRegTop[n] and PackedRegHeight[n] shall be less than or equal to `packed_picture_height` / VerDiv1. Otherwise, the sum of PackedRegTop[n] and PackedRegHeight[n] shall be less than or equal to `packed_picture_height` / VerDiv1 * 2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- When the decoded picture has 4:2:0 or 4:2:2 chroma format, PackedRegLeft[n] shall correspond to an even horizontal coordinate value of luma sample units, and PackedRegWidth[n] shall correspond to an even number of luma samples, both within the decoded picture.
- When the decoded picture has 4:2:0 chroma format, PackedRegTop[n] shall correspond to an even vertical coordinate value of luma sample units, and ProjRegHeight[n] shall correspond to an even number of luma samples, both within the decoded picture.

7.5.4 Rotation structure

7.5.4.1 Definition

The fields in this structure provides the yaw, pitch, and roll angles, respectively, of the rotation to be applied to convert the local coordinate axes to the global coordinate axes. In the case of stereoscopic omnidirectional video, the fields apply to each view individually.

7.5.4.2 Syntax

```
aligned(8) class RotationStruct() {
   signed int(32) rotation_yaw;
   signed int(32) rotation_pitch;
   signed int(32) rotation_roll;
}
```

7.5.4.3 Semantics

`rotation_yaw`, `rotation_pitch`, and `rotation_roll` specify the yaw, pitch, and roll angles, respectively, of the rotation that is applied to the unit sphere to convert the local coordinate axes to the global coordinate axes, in units of 2^{-16} degrees, relative to the global coordinate axes. `rotation_yaw` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. `rotation_pitch` shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive. `rotation_roll` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive.

7.5.5 Content coverage structure

7.5.5.1 Definition

The fields in this structure provides the content coverage, which is expressed by one or more sphere regions covered by the content, relative to the global coordinate axes.

7.5.5.2 Syntax

```
aligned(8) class ContentCoverageStruct() {
   unsigned int(8) coverage_shape_type;
   unsigned int(8) num_regions;
   unsigned int(1) view_idc_presence_flag;
   if (view_idc_presence_flag == 0) {
      unsigned int(2) default_view_idc;
      bit(5) reserved = 0;
   } else
      bit(7) reserved = 0;
   for ( i = 0; i < num_regions; i++) {
      if (view_idc_presence_flag == 1) {
         unsigned int(2) view_idc[i];
         bit(6) reserved = 0;
      }
      SphereRegionStruct(1);
   }
}
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.5.5.3 Semantics

`coverage_shape_type` specifies the shape of the sphere regions expressing the content coverage. `coverage_shape_type` has the same semantics as `shape_type` specified in subclause 7.7.2.3. The value of `coverage_shape_type` is used as the shape type value when applying subclause 7.5.6 to the semantics of `ContentCoverageStruct`.

`num_regions` specifies the number of sphere regions. Value 0 is reserved.

`view_idc_presence_flag` equal to 0 specifies that `view_idc[i]` is not present. `view_idc_presence_flag` equal to 1 specifies that `view_idc[i]` is present and indicates the association of sphere regions with particular (left, right, or both) views.

`default_view_idc` equal to 0 indicates that each sphere region is monoscopic, 1 indicates that each sphere region is on the left view of a stereoscopic content, 2 indicates that each sphere region is on the right view of a stereoscopic content, 3 indicates that each sphere region is on both the left and right views.

`view_idc[i]` equal to 1 indicates that the i-th sphere region is on the left view of a stereoscopic content, 2 indicates the i-th sphere region is on the right view of a stereoscopic content, and 3 indicates that the i-th sphere region is on both the left and right views. `view_idc[i]` equal to 0 is reserved.

NOTE `view_idc_presence_flag` equal to 1 enables indicating asymmetric stereoscopic coverage. For example, one example of an asymmetric stereoscopic coverage could be described by setting `num_regions` equal to 2, indicating one sphere region to be on the left view covering the azimuth range of −90° to 90°, inclusive, and indicating the other sphere region to be on the right view covering the azimuth range of −60 to 60°, inclusive.

When `SphereRegionStruct(1)` is included in the `ContentCoverageStruct()`, subclause 7.5.6 applies and `interpolate` shall be equal to 0.

The content coverage is specified by the union of `num_regions SphereRegionStruct(1)` structure(s). When `num_regions` is greater than 1, the content coverage may be non-contiguous.

7.5.6 Sphere region structure

7.5.6.1 Definition

The sphere region structure (`SphereRegionStruct`) specifies a sphere region.

When `centre_tilt` is equal to 0, the sphere region specified by this structure is derived as follows:

- If both `azimuth_range` and `elevation_range` are equal to 0, the sphere region specified by this structure is a point on a spherical surface.
- Otherwise, the sphere region is defined using variables centreAzimuth, centreElevation, cAzimuth1, cAzimuth, cElevation1, and cElevation2 derived as follows:

centreAzimuth = `centre_azimuth` ÷ 65536
centreElevation = `centre_elevation` ÷ 65536
cAzimuth1 = (`centre_azimuth` – `azimuth_range` ÷ 2) ÷ 65536
cAzimuth2 = (`centre_azimuth` + `azimuth_range` ÷ 2) ÷ 65536
cElevation1 = (`centre_elevation` – `elevation_range` ÷ 2) ÷ 65536
cElevation2 = (`centre_elevation` + `elevation_range` ÷ 2) ÷ 65536

The sphere region is defined as follows with reference to the shape type value specified in the semantics of the structure containing this instance of `SphereRegionStruct`:

- When the shape type value is equal to 0, the sphere region is specified by four great circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in Figure 20.
- When the shape type value is equal to 1, the sphere region is specified by two azimuth circles and two elevation circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in Figure 21.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

When `centre_tilt` is not equal to 0, the sphere region is firstly derived as above and then a tilt rotation is applied 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. The final sphere region is the one after applying the tilt rotation.

Shape type value equal to 0 specifies that the sphere region is specified by four great circles as illustrated in Figure 20.

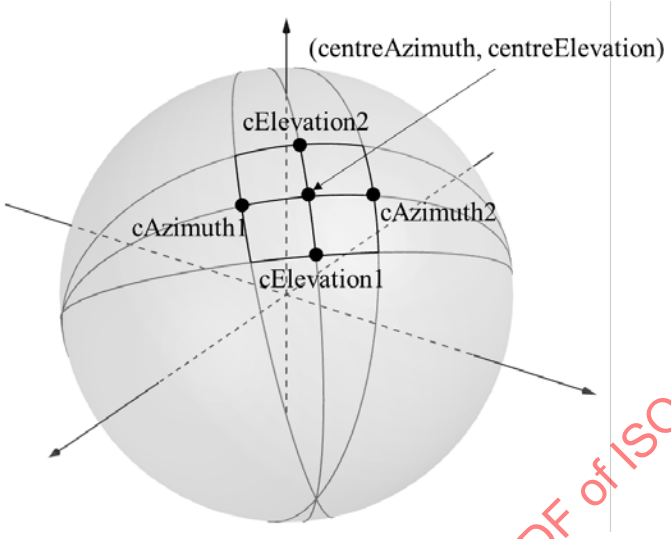


Figure 19 – A sphere region specified by four great circles

Shape type value equal to 1 specifies that the sphere region is specified by two azimuth circles and two elevation circles as illustrated in Figure 21.

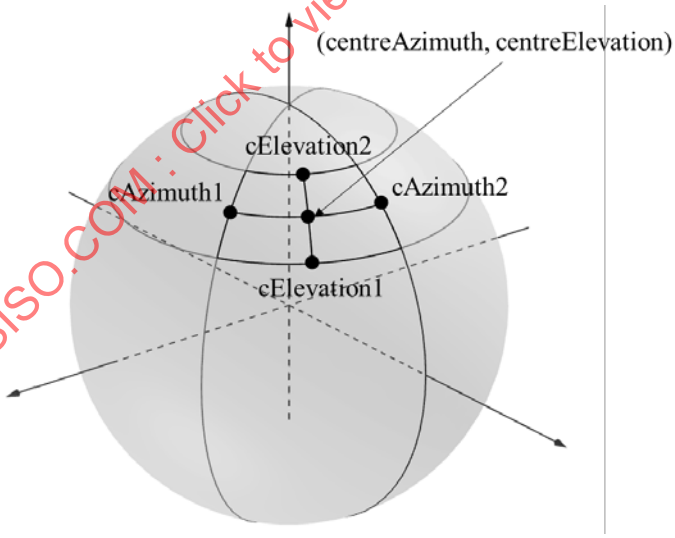


Figure 20 – A sphere region specified by two azimuth circles and two elevation circles

Shape type values greater than 1 are reserved.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.5.6.2 Syntax

```
aligned(8) SphereRegionStruct(range_included_flag) {
   signed int(32) centre_azimuth;
   signed int(32) centre_elevation;
   signed int(32) centre_tilt;
   if (range_included_flag) {
      unsigned int(32) azimuth_range;
      unsigned int(32) elevation_range;
   }
   unsigned int(1) interpolate;
   bit(7) reserved = 0;
}
```

7.5.6.3 Semantics

`centre_azimuth` and `centre_elevation` specify the azimuth and elevation values, respectively, of the centre of the sphere region in units of 2^{-16} degrees. `centre_azimuth` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. `centre_elevation` shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive.

`centre_tilt` specifies the tilt angle of the sphere region in units of 2^{-16} degrees. `centre_tilt` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive.

`azimuth_range` and `elevation_range`, when present, specify the azimuth and elevation ranges, respectively, of the sphere region specified by this structure in units of 2^{-16} degrees. `azimuth_range` and `elevation_range` specify the range through the centre point of the sphere region, as illustrated by Figure 19 or Figure 20. When `azimuth_range` and `elevation_range` are not present in this instance of `SphereRegionStruct`, they are inferred as specified in the semantics of the structure containing this instance of `SphereRegionStruct`. `azimuth_range` shall be in the range of 0 to $360 * 2^{16}$, inclusive. `elevation_range` shall be in the range of 0 to $180 * 2^{16}$, inclusive.

The semantics of `interpolate` are specified by the semantics of the structure containing this instance of `SphereRegionStruct`.

7.6 Restricted video schemes for omnidirectional video

7.6.1 Scheme types

7.6.1.1 Open-ended and closed scheme types

An open-ended scheme type for a kind of transformation is a scheme type that allows future extensions. For example, the `'stvi'` scheme type (stereoscopic video) may be used for a new frame packing arrangement type that is defined after the definition of the `'stvi'` scheme type. A closed scheme type, on the other hand, when defined, clearly specifies which transformations are allowed and does not allow future extensions.

`SchemeTypeBox` allows inclusion of only one scheme type and there may be only one instance of `SchemeTypeBox` included in `RestrictedSchemeInfoBox`. The scheme type included in `SchemeTypeBox` shall be an open-ended scheme type, i.e., `'podv'` or `'fodv'`.

`CompatibleSchemeTypeBox` allows inclusion of only one scheme type but there may be multiple instances of `CompatibleSchemeTypeBox` included in `RestrictedSchemeInfoBox`. A closed scheme type shall only be included in an instance of `CompatibleSchemeTypeBox`.

An OMAF player that does not recognize the open-ended scheme type in `SchemeTypeBox` shall ignore the track. An OMAF player that recognizes the open-ended scheme type in `SchemeTypeBox` but none of the scheme types in the instances of `CompatibleSchemeTypeBox`, if any, should parse all boxes contained in `SchemeInformationBox` to determine whether it has the capability required to properly process the track. An OMAF player should ignore the track unless it supports all boxes contained in `SchemeInformationBox` and all syntax elements and syntax element values present in those boxes.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.6.1.2 Projected omnidirectional video ('podv')

The use of the projected omnidirectional video scheme for the restricted video sample entry type 'resv' indicates that the decoded pictures are packed pictures containing either monoscopic or stereoscopic content. The use of the projected omnidirectional video scheme is indicated by scheme_type equal to 'podv' (projected omnidirectional video) within SchemeTypeBox in the RestrictedSchemeInfoBox.

The format of the projected monoscopic pictures is indicated with the ProjectedOmniVideoBox contained within the SchemeInformationBox. One and only one ProjectedOmniVideoBox shall be present in the SchemeInformationBox when the scheme type is 'podv'.

The 'podv' scheme type is defined as an open-ended scheme type for projected omnidirectional video.

As specified in subclause 7.6.2, a ProjectionFormatBox shall be present within the ProjectedOmniVideoBox. ProjectionFormatBox is not constrained beyond the specification in subclause 7.6.2. The 'podv' scheme type may be used with all version values specified for ProjectionFormatBox in this document and in any of its future amendments and editions. The 'podv' scheme type may be used with any projection_type value specified in this document and in any of its future amendments and editions.

When the ProjectedOmniVideoBox is present in the SchemeInformationBox, StereoVideoBox may be present in the same SchemeInformationBox.

For stereoscopic video, the frame packing arrangement of the projected left and right pictures is indicated with the StereoVideoBox contained within the SchemeInformationBox. The absence of StereoVideoBox indicates that the omnidirectionally projected content of the track is monoscopic. When StereoVideoBox is present in the SchemeInformationBox for the omnidirectional video scheme, version shall be equal to 0, stereo_scheme shall be equal to 4 and the first byte of stereo_indication_type shall be equal to 3, 4, or 5 indicating that the top-bottom frame packing, the side-by-side frame packing, or the temporal interleaving of alternating first and second constituent frames, respectively, is in use and the second byte of stereo_indication_type shall be equal to 0 indicating that quincunx sampling is not in use.

NOTE The 'stvi' scheme type is not expected to be used when the 'podv' scheme type is used.

Optional region-wise packing is indicated with the RegionWisePackingBox contained within the ProjectedOmniVideoBox. The absence of RegionWisePackingBox indicates that no region-wise packing is applied, i.e., that the packed picture is identical to the projected picture.

RegionWisePackingBox is not constrained beyond the specification in subclause 7.6.4. The 'podv' scheme type may be used with all version values specified for RegionWisePackingBox in this document and in any of its future amendments and editions. The 'podv' scheme type may be used with any values of the syntax elements of RegionWisePackingBox specified and allowed in this document and in any of its future amendments and editions.

In addition to the boxes constrained above, SchemeInformationBox may directly or indirectly contain any boxes allowed by this document and in any of its future amendments and editions. Those boxes are not constrained beyond their definition, syntax, and semantics.

7.6.1.3 Equirectangular projected video ('erpv')

NOTE This scheme type could be used for specifying media profiles.

The 'erpv' scheme type is defined as a closed scheme type for projected omnidirectional video.

When scheme_type is equal to 'erpv' in an instance of CompatibleSchemeTypeBox in the RestrictedSchemeInfoBox, the track conforms to the constraints of scheme_type equal to 'podv' with all of the following additional constraints:

- ProjectionFormatBox within the ProjectedOmniVideoBox shall indicate the equirectangular projection.
- When RegionWisePackingBox is present, the following constraints all apply:
  - The value of NumRegions shall be equal to HorDiv1 * VerDiv1.
  - For each value of i in the range of 0 to NumRegions − 1, inclusive, the following applies:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

  - The value of PackingType[i] shall be equal to 0.
  - The value of TransformType[i] shall be equal to 0.
  - The value of PackedRegWidth[i] shall be equal to ProjRegWidth[i].
  - The value of PackedRegHeight[i] shall be equal to ProjRegHeight[i].
- `version` of `ProjectionFormatBox`, `StereoVideoBox` (when present), `RegionWisePackingBox` (when present), `RotationBox` (when present), and `CoverageInformationBox` (when present) shall be equal to 0.
- `SchemeInformationBox` shall not directly or indirectly contain any boxes other than `ProjectedOmniVideoBox`, `ProjectionFormatBox`, `StereoVideoBox`, `RegionWisePackingBox`, `RotationBox`, and `CoverageInformationBox`.

7.6.1.4 Packed equirectangular or cubemap projected video (`'ercm'`)

NOTE This scheme type could be used for specifying media profiles.

The `'ercm'` scheme type is defined as a closed scheme type for projected omnidirectional video.

When `scheme_type` is equal to `'ercm'` in an instance of `CompatibleSchemeTypeBox` in the `RestrictedSchemeInfoBox`, the track conforms to the constraints of `scheme_type` equal to `'podv'`, `scheme_type` equal to `'podv'` shall be present in `SchemeTypeBox` in the `RestrictedSchemeInfoBox`, and all of the following additional constraints apply:

- `ProjectionFormatBox` within the `ProjectedOmniVideoBox` shall indicate either the equirectangular projection or the cubemap projection.
- When `RegionWisePackingBox` is present, the value of `packing_type[i]` for each value of i shall be equal to 0.
- `version` of `ProjectionFormatBox`, `StereoVideoBox` (when present), `RegionWisePackingBox` (when present), `RotationBox` (when present), and `CoverageInformationBox` (when present) shall be equal to 0.
- `SchemeInformationBox` shall not directly or indirectly contain any boxes other than `ProjectedOmniVideoBox`, `ProjectionFormatBox`, `StereoVideoBox`, `RegionWisePackingBox`, `RotationBox`, and `CoverageInformationBox`.

7.6.1.5 Fisheye omnidirectional video (`'fodv'`)

The use of the fisheye omnidirectional video scheme for the restricted video sample entry type `'resv'` indicates that the decoded pictures are fisheye video pictures. The use of the fisheye omnidirectional video scheme is indicated by `scheme_type` equal to `'fodv'` (fisheye omnidirectional video) within `SchemeTypeBox` in the `RestrictedSchemeInfoBox`.

The format of fisheye video is indicated with the `FisheyeOmniVideoBox` contained within the `SchemeInformationBox`. One and only one `FisheyeOmniVideoBox` shall be present in the `SchemeInformationBox` when the scheme type is `'fodv'` within `SchemeTypeBox` in the `RestrictedSchemeInfoBox`.

The `'fodv'` scheme type is defined as an open-ended scheme type for fisheye omnidirectional video.

The `'fodv'` scheme type may be used with all `version` values specified for `FisheyeVideoEssentialInfoBox` and `FisheyeVideoSupplementalInfoBox` in this document and in any of its future amendments and editions. The `'fodv'` scheme type may be used with any values of syntax elements of `FisheyeOmniVideoBox` allowed in this document and in any of its future amendments and editions.

When `FisheyeOmniVideoBox` is present in the `SchemeInformationBox`, `StereoVideoBox` shall not be present in the same `SchemeInformationBox`.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

In addition to the boxes constrained above, `SchemeInformationBox` may directly or indirectly contain any boxes allowed by this document and in any of its future amendments and editions. Those boxes are not constrained beyond their definition, syntax, and semantics.

7.6.2 Projected omnidirectional video box

7.6.2.1 Definition

Box Type: `'povd'`
Container: `SchemeInformationBox`
Mandatory: Yes, when `scheme_type` is equal to `'podv'`
Quantity: Zero or one

This box is a container box that contains boxes indicating information for the following:

- the projection format of the projected picture (C for monoscopic video contained in the track, C_L and C_R for left and right view of stereoscopic video),
- region-wise packing, when applicable,
- the rotation for conversion between the local coordinate axes and the global coordinate axes, if applied, and
- optionally the content coverage of the track.

The values of the variables HorDiv1 and VerDiv1 are set as follows:

- If `StereoVideoBox` is not present in `SchemeInformationBox`, HorDiv1 is set equal to 1 and VerDiv1 is set equal to 1.
- Otherwise (`StereoVideoBox` is present in `SchemeInformationBox`), the following applies:
  - If side-by-side frame packing is indicated, HorDiv1 is set equal to 2 and VerDiv1 is set equal to 1.
  - Otherwise if top-bottom frame packing is indicated, HorDiv1 is set equal to 1 and VerDiv1 is set equal to 2.
  - Otherwise (temporal interleaving is indicated), HorDiv1 and VerDiv1 are both set equal to 1.

If `RotationBox` is not present in `ProjectedOmniVideoBox`, RotationFlag is set equal to 0. Otherwise, RotationFlag is set equal to 1.

If `StereoVideoBox` is not present in `SchemeInformationBox`, SpatiallyPackedStereoFlag, TopBottomFlag, and SideBySideFlag are set equal to 0. Otherwise, the following applies:

- When the `StereoVideoBox` indicates top-bottom frame packing, SpatiallyPackedStereoFlag is set equal to 1, TopBottomFlag is set equal to 1, and SideBySideFlag is set equal to 0.
- When the `StereoVideoBox` indicates side-by-side frame packing, SpatiallyPackedStereoFlag is set equal to 1, TopBottomFlag is set equal to 0, and SideBySideFlag is set equal to 1.
- When the `StereoVideoBox` indicates temporal interleaving, SpatiallyPackedStereoFlag, TopBottomFlag, and SideBySideFlag are all set equal to 0.

The following applies:

- The width and height of a monoscopic projected luma picture (ConstituentPicWidth and ConstituentPicHeight, respectively) are derived as follows:
  - If `RegionWisePackingBox` is not present in `ProjectedOmniVideoBox`, ConstituentPicWidth and ConstituentPicHeight are set to be equal to `width` / HorDiv1 and `height` / VerDiv1, respectively, where `width` and `height` are syntax elements of `VisualSampleEntry`.
  - Otherwise, ConstituentPicWidth and ConstituentPicHeight are set equal to `proj_picture_width` / HorDiv1 and `proj_picture_height` / VerDiv1, respectively.
- If `RegionWisePackingBox` is not present in `ProjectedOmniVideoBox`, RegionWisePackingFlag is set equal to 0. Otherwise, RegionWisePackingFlag is set equal to 1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

– The semantics of the sample locations of each decoded picture resulting by decoding the samples referring to this sample entry are specified in subclause 7.5.1.2.

7.6.2.2 Syntax

```
aligned(8) class ProjectedOmniVideoBox extends Box('povd') {
   ProjectionFormatBox(); // mandatory
   // optional boxes but no fields
}

aligned(8) class ProjectionFormatBox() extends FullBox('prfr', 0, 0) {
   ProjectionFormatStruct();
}
```

7.6.3 Fisheye omnidirectional video box

7.6.3.1 Definition

Box Type: `'fovd'`
Container: `SchemeInformationBox`
Mandatory: Yes, when `scheme_type` is equal to `'fodv'`
Quantity: Zero or one

`FisheyeOmniVideoBox` provides essential fisheye video parameters for stitching and rendering of fisheye video at the OMAF player. The fields in `FisheyeOmniVideoBox` provide region information of circular images in the coded picture and field of view and camera parameters of fisheye lens. `FisheyeVideoSupplementalInfoBox` provides the local field of view information for high quality stitching and rendering of fisheye video, as well as deadzone information.

7.6.3.2 Syntax

```
aligned(8) class FisheyeOmniVideoBox extends Box('fovd') {
   FisheyeVideoEssentialInfoBox(); // mandatory
   FisheyeVideoSupplementalInfoBox(); // optional
}

aligned(8) class FisheyeVideoEssentialInfoBox extends FullBox('fovi', 0, 0) {
   FisheyeVideoEssentialInfoStruct();
}

aligned(8) class FisheyeVideoSupplementalInfoBox extends FullBox('fvsi', 0, 0) {
   FisheyeVideoSupplementalInfoStruct();
}
```

7.6.4 Region-wise packing box

7.6.4.1 Definition

Box Type: `'rwpk'`
Container: `ProjectedOmniVideoBox`
Mandatory: No
Quantity: Zero or one

`RegionWisePackingBox` specifies the mapping between packed regions and the corresponding projected regions and specifies the location and size of the guard bands, if any.

NOTE Among other information the `RegionWisePackingBox` also provides the content coverage information in the 2D Cartesian picture domain.

7.6.4.2 Syntax

```
aligned(8) class RegionWisePackingBox extends FullBox('rwpk', 0, 0) {
   RegionWisePackingStruct();
}
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.6.4.3 Semantics

Subclause 7.5.3 applies with the following additional constraint:

- `packed_picture_width` and `packed_picture_height` shall have such values that `packed_picture_width` is an integer multiple of `width` and `packed_picture_height` is an integer multiple of `height`, where `width` and `height` are syntax elements of the `VisualSampleEntry` containing this box.

7.6.5 Rotation box

7.6.5.1 Definition

Box Type: `'rotn'`
Container: `ProjectedOmniVideoBox`
Mandatory: No
Quantity: Zero or one

The fields in this box provides the yaw, pitch, and roll angles, respectively, of the rotation to be applied to convert the local coordinate axes to the global coordinate axes. In the case of stereoscopic omnidirectional video, the fields apply to each view individually. When the `RotationBox` is not present, the fields `rotation_yaw`, `rotation_pitch`, and `rotation_roll` are all inferred to be equal to 0.

7.6.5.2 Syntax

```
aligned(8) class RotationBox extends FullBox('rotn', 0, 0) {
   RotationStruct();
}
```

7.6.6 Coverage information box

7.6.6.1 Definition

Box Type: `'covi'`
Container: `ProjectedOmniVideoBox`
Mandatory: No
Quantity: Zero or one

This box provides information on the content coverage of this track.

NOTE It is totally up to the OMAF player to handle the area that is not covered by the content when rendering the omnidirectional video content.

Each sphere location within the sphere regions specifying the content coverage shall have a corresponding sample in the decoded pictures. However, there may be some sphere locations that do have corresponding samples in the decoded pictures but are outside the content coverage.

7.6.6.2 Syntax

```
aligned(8) class CoverageInformationBox extends FullBox('covi', 0, 0) {
   ContentCoverageStruct()
}
```

7.7 Timed metadata for sphere regions

7.7.1 General

Subclause 7.7 specifies a generic timed metadata track syntax for indicating sphere regions. The purpose for the timed metadata track is indicated by the track sample entry type. The sample format of all metadata tracks specified in subclause 7.7 starts with a common part and may be followed by an extension part that is specific to the sample entry of the metadata track. Each sample specifies a sphere region.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

When a sphere region timed metadata track is linked to one or more media tracks with a 'cdsc' track reference, it describes each media track individually.

NOTE The syntax allows for one sample to specify multiple sphere regions. However, there is a semantic restriction that limits the samples to have only one sphere region.

7.7.2 Sample entry

7.7.2.1 Definition

Exactly one SphereRegionConfigBox shall be present in the sample entry. SphereRegionConfigBox specifies the shape of the sphere region specified by the samples. When the azimuth and elevation ranges of the sphere region in the samples do not change, they may be indicated in the sample entry.

7.7.2.2 Syntax

```
class SphereRegionSampleEntry(type) extends MetaDataSampleEntry(type) {
   SphereRegionConfigBox(); // mandatory
   Box[] other_boxes; // optional
}

class SphereRegionConfigBox extends FullBox('rosc', 0, 0) {
   unsigned int(8) shape_type;
   bit(7) reserved = 0;
   unsigned int(1) dynamic_range_flag;
   if (dynamic_range_flag == 0) {
      unsigned int(32) static_azimuth_range;
      unsigned int(32) static_elevation_range;
   }
   unsigned int(8) num_regions;
}
```

7.7.2.3 Semantics

shape_type equal to 0 specifies that the sphere region is specified by four great circles. shape_type equal to 1 specifies that the sphere region is specified by two azimuth circles and two elevation circles. shape_type values greater than 1 are reserved. The value of shape_type is used as the shape type value when applying clause 7.5.6 to the semantics of the samples of the sphere region metadata track.

dynamic_range_flag equal to 0 specifies that the azimuth and elevation ranges of the sphere region remain unchanged in all samples referring to this sample entry. dynamic_range_flag equal to 1 specifies that the azimuth and elevation ranges of the sphere region are indicated in the sample format.

static_azimuth_range and static_elevation_range specify the azimuth and elevation ranges, respectively, of the sphere region for each sample referring to this sample entry in units of 2^{-16} degrees. static_azimuth_range and static_elevation_range specify the ranges through the centre point of the sphere region, as illustrated by Figure 19 or Figure 20. static_azimuth_range shall be in the range of 0 to $360 * 2^{16}$, inclusive. static_elevation_range shall be in the range of 0 to $180 * 2^{16}$, inclusive. When static_azimuth_range and static_elevation_range are present and are both equal to 0, the sphere region for each sample referring to this sample entry is a point on a spherical surface. When static_azimuth_range and static_elevation_range are present, the values of azimuth_range and elevation_range are inferred to be equal to static_azimuth_range and static_elevation_range, respectively, when applying subclause 7.5.6 to the semantics of the samples of the sphere region metadata track.

num_regions specifies the number of sphere regions in the samples referring to this sample entry. num_regions shall be equal to 1. Other values of num_regions are reserved.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.7.3 Sample format

7.7.3.1 Definition

Each sample specifies a sphere region. The `SphereRegionSample` structure may be extended in derived track formats.

7.7.3.2 Syntax

```
aligned(8) SphereRegionSample() {
   for (i = 0; i < num_regions; i++)
      SphereRegionStruct(dynamic_range_flag);
}
```

7.7.3.3 Semantics

Subclause 7.5.6 applies to the sample that contains the `SphereRegionStruct` structure.

Let the target media samples be the media samples in the referenced media tracks with composition times greater than or equal to the composition time of this sample and less than the composition time of the next sample.

`interpolate` equal to 0 specifies that the values of `centre_azimuth`, `centre_elevation`, `centre_tilt`, `azimuth_range` (if present), and `elevation_range` (if present) in this sample apply to the target media samples. `interpolate` equal to 1 specifies that the values of `centre_azimuth`, `centre_elevation`, `centre_tilt`, `azimuth_range` (if present), and `elevation_range` (if present) that apply to the target media samples are linearly interpolated from the values of the corresponding fields in this sample and the previous sample.

The value of `interpolate` for a sync sample, the first sample of the track, and the first sample of a track fragment shall be equal to 0.

7.7.4 Initial viewing orientation

7.7.4.1 Definition

This metadata indicates initial viewing orientations that should be used when playing the associated media tracks or a single omnidirectional image stored as an image item. In the absence of this type of metadata `centre_azimuth`, `centre_elevation`, and `centre_tilt` should all be inferred to be equal to 0.

An OMAF player should use the indicated or inferred `centre_azimuth`, `centre_elevation`, and `centre_tilt` values as follows:

- If the orientation/viewport metadata of the OMAF player is obtained on the basis of an orientation sensor included in or attached to a viewing device, the OMAF player should
  - obey only the `centre_azimuth` value, and
  - ignore the values of `centre_elevation` and `centre_tilt` and use the respective values from the orientation sensor instead.
- Otherwise, the OMAF player should obey all three of `centre_azimuth`, `centre_elevation`, and `centre_tilt`.

The track sample entry type `'invo'` shall be used.

`shape_type` shall be equal to 0, `dynamic_range_flag` shall be equal to 0, `static_azimuth_range` shall be equal to 0, and `static_elevation_range` shall be equal to 0 in the `SphereRegionConfigBox` of the sample entry.

NOTE This metadata applies to any viewport regardless of which azimuth and elevation ranges are covered by the viewport. Thus, `dynamic_range_flag`, `static_azimuth_range`, and `static_elevation_range` do not affect the dimensions of the viewport that this metadata concerns and are hence required to be equal to 0. When the OMAF player obeys the `centre_tilt` value as concluded above, the value of `centre_tilt` could be interpreted by setting the azimuth and elevation ranges for the sphere region of the viewport equal to those that are actually used in displaying the viewport.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

7.7.4.2 Sample syntax

```
class InitialViewingOrientationSample() extends SphereRegionSample() {
   unsigned int(1) refresh_flag;
   bit(7) reserved = 0;
}
```

7.7.4.3 Sample semantics

NOTE 1 As the sample structure extends from `SphereRegionSample`, the syntax elements of `SphereRegionSample` are included in the sample.

`centre_azimuth`, `centre_elevation`, and `centre_tilt` specify the viewing orientation in units of 2^{-16} degrees relative to the global coordinate axes. `centre_azimuth` and `centre_elevation` indicate the centre of the viewport, and `centre_tilt` indicates the tilt angle of the viewport.

`interpolate` shall be equal to 0.

`refresh_flag` equal to 0 specifies that the indicated viewing orientation should be used when starting the playback from a time-parallel sample in an associated media track. `refresh_flag` equal to 1 specifies that the indicated viewing orientation should always be used when rendering the time-parallel sample of each associated media track, i.e., both in continuous playback and when starting the playback from the time-parallel sample.

NOTE 2 `refresh_flag` equal to 1 enables the content author to indicate that a particular viewing orientation is recommended even when playing the video continuously. For example, `refresh_flag` equal to 1 could be indicated for a scene cut position.

7.7.5 Recommended viewport

The recommended viewport timed metadata track indicates the viewport that should be displayed when the user does not have control of the viewing orientation or has released control of the viewing orientation.

NOTE The recommended viewport timed metadata track could be used for indicating a recommended viewport based on a director's cut or based on measurements of viewing statistics.

The track sample entry type `'rcvp'` shall be used.

The sample entry of this sample entry type is specified as follows:

```
class RcvpSampleEntry() extends SphereRegionSampleEntry('rcvp') {
   RcvpInfoBox(); // mandatory
}

class RcvpInfoBox extends FullBox('rvif', 0, 0) {
   unsigned int(8) viewport_type;
   string viewport_description;
}
```

`viewport_type` specifies the type of the recommended viewport as listed in Table 11.

Table 11 – Recommended viewport type

Value	Description
0	A recommended viewport per the director's cut, i.e., a viewport suggested according to the creative intent of the content author or content provider
1	A recommended viewport selected based on measurements of viewing statistics
2..239	Reserved (for use by future extensions of ISO/IEC 23090-2)
240..255	Unspecified (for use by applications or external specifications)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`viewport_description` is null-terminated UTF-8 string that provides a textual description of the recommended viewport.

The sample syntax of `SphereRegionSample` shall be used.

`shape_type` shall be equal to 0 in the `SphereRegionConfigBox` of the sample entry.

`static_azimuth_range` and `static_elevation_range`, when present, or `azimuth_range` and `elevation_range`, when present, indicate the azimuth and elevation ranges, respectively, of the recommended viewport.

`centre_azimuth` and `centre_elevation` indicate the centre point of the recommended viewport relative to the global coordinate axes. `centre_tilt` indicates the tilt angle of the recommended viewport.

7.7.6 Timed text sphere location metadata

7.7.6.1 General

The timed text sphere location metadata indicates the timed text sphere lcation that is used, together with other information, to determine where the timed text is placed and displayed in 3D space. Since the timed text cues are rendered at certain positions in 3D space, they are only visible when timed text sphere location is within the sphere region defining the viewport.

7.7.6.2 Sample entry format

The track sample entry type `'ttsl'` shall be used.

The sample entry of this sample entry type is specified as follows:

```
class TTSphereLocationSampleEntry() extends SphereRegionSampleEntry('ttsl') {
   unsigned int(1) depth_included_flag;
   bit(7) reserved = 0;
}
```

`depth_included_flag` equal to 1 specifies that the depth (z-value) of the regions on which the timed text is to be rendered is present in the samples. The value 0 specifies that the depth (z-value) of the regions on which the timed text is to be rendered is not present in the samples.

When `SphereRegionSampleEntry()` is included in `TTSphereLocationSampleEntry()`, the following applies:

The values of `shape_type`, `dynamic_range_flag`, `static_azimuth_range`, and `static_elevation_range` shall all be equal to 0.

The value of `num_regions` may be greater than 1.

7.7.6.3 Sample format

The sample format of timed text sphere location timed metadata is specified as follows:

```
aligned(8) TTSphereLocationSample() extends SphereRegionSample() {
   for (i=0; i<num_regions; i++) {
      string region_id;
      if (depth_included_flag)
         unsigned int(16) region_depth;
   }
}
```

`region_id` provides the identifier of the region on which the timed text is to be rendered. `region_id` should be equal to the identification of the corresponding region defined in the timed text streams in the IMSC1 or WebVTT track.

`region_depth` indicates the depth (z-value) of the region on which the timed text is to be rendered. The depth value is the norm of the normal vector of the timed text region. This value is relative to a unit sphere and is in units of 2^{-16}.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

When SphereRegionStruct() is included in the TTSphereLocationSample() structure, it indicates the timed text sphere location that is used, together with other information, to determine where the timed text is placed and displayed in 3D space.

7.8 Signalling of region-wise quality ranking

7.8.1 General

Quality ranking values of quality ranking regions relative to other quality ranking regions of the same track or quality ranking regions of other tracks may be indicated by using the SphereRegionQualityRankingBox or the 2DRegionQualityRankingBox. When neither SphereRegionQualityRankingBox nor 2DRegionQualityRankingBox is present in a visual sample entry, the quality ranking value for the visual track is not defined. Quality ranking values indicate a relative quality order of quality ranking regions. When quality ranking region A has a non-zero quality ranking value less than that of quality ranking region B, quality ranking region A has a higher quality than quality ranking region B. When the quality ranking value is non-zero, the picture quality within the entire indicated quality ranking region is approximately constant. The boundaries of the quality ranking sphere regions specified by the SphereRegionQualityRankingBox may or may not match with the boundaries of the quality ranking 2D regions specified by the 2DRegionQualityRankingBox. The boundaries of the quality ranking sphere or 2D regions may or may not match with the boundaries of the packed regions or the boundaries of the projected regions specified by RegionWisePackingBox.

7.8.2 Spherical region-wise quality ranking

7.8.2.1 Definition

Box type:	'srqr'
Container:	VisualSampleEntry
Mandatory (per an item):	No
Quantity (per an item):	At most one for each region_definition_type value

7.8.2.2 Syntax

```
aligned(8) class SphereRegionQualityRankingBox extends FullBox('srqr', 0, 0) {
   SphereRegionQualityRankingStruct();
}

aligned(8) class SphereRegionQualityRankingStruct() {
   unsigned int(8) region_definition_type;
   unsigned int(8) num_regions;
   unsigned int(1) remaining_area_flag;
   unsigned int(1) view_idc_presence_flag;
   unsigned int(1) quality_ranking_local_flag;
   unsigned int(4) quality_type;
   bit(1) reserved = 0;
   if (view_idc_presence_flag == 0) {
      unsigned int(2) default_view_idc;
      bit(6) reserved = 0;
   }
   for (i = 0; i < num_regions; i++) {
      unsigned int(8) quality_ranking;
      if (view_idc_presence_flag == 1) {
         unsigned int(2) view_idc;
         bit(6) reserved = 0;
      }
      if (quality_type == 1) {
         unsigned int(16) orig_width;
         unsigned int(16) orig_height;
      }
      if ((i < (num_regions - 1)) || (remaining_area_flag == 0))
         SphereRegionStruct(1);
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
    }
}
```

Semantics

`region_definition_type` has identical semantics to `shape_type` of `SphereRegionConfigBox`.

`num_regions` specifies the number of quality ranking sphere regions for which the quality ranking information is given in this box. Value 0 is reserved. There shall be no point on the sphere that is contained in more than one of these quality ranking sphere regions.

`remaining_area_flag` equal to 0 specifies that all the quality ranking sphere regions are defined by the `SphereRegionStruct(1)` structures. `remaining_area_flag` equal to 1 specifies that the first `num_regions − 1` quality ranking sphere regions are defined by `SphereRegionStruct(1)` structure and the last remaining quality ranking sphere region is the sphere region within the content coverage, not covered by the union of the quality ranking sphere regions defined by the first `num_regions − 1` `SphereRegionStruct(1)` structures. The last remaining quality ranking sphere region may be on both the left and right views.

`view_idc_presence_flag` equal to 0 specifies that `view_idc` is not present. `view_idc_presence_flag` equal to 1 specifies that `view_idc` is present and indicates the association of quality ranking sphere region with particular (left or right or both) views or monoscopic content.

`quality_ranking_local_flag` equal to 1 specifies that the quality ranking information provided in this instance of this box is relative to the quality ranking information provided in all instances of this box for this track only. `quality_ranking_local_flag` equal to 0 specifies that the quality ranking information provided in this instance of this box is relative to the quality ranking information provided in all instances of `SphereRegionQualityRankingBox` and `2DRegionQualityRankingBox` with `quality_ranking_local_flag` equal to 0 for this and other tracks.

NOTE 1 `quality_ranking_local_flag` equal to 1 could be used in a track that includes coded video data by reference to a set of tracks among which one is selected as the source of the coded video data. For example, an `'alte'` track reference specified in subclause 7.2.1 could be used for this purpose. `SphereRegionQualityRankingBox` or `2DQualityRankingBox` with `quality_ranking_local_flag` equal to 0 could be present in the referenced tracks within the file.

`quality_type` indicates which factor causes the differences in the quality of packed regions on the picture. `quality_type` equal to 0 specifies that all packed regions correspond to the same projected picture resolution. `quality_type` equal to 1 specifies that at least one horRatio value, as derived in subclause 5.4.2, may differ from other horRatio values among all pairs of packed and projected regions of the picture or at least one verRatio value, as derived in subclause 5.4.2, may differ from other verRatio values among all pairs of packed and projected regions of the picture. `quality_type` values greater than 1 are reserved.

`default_view_idc` equal to 0 indicates that the quality ranking sphere region is monoscopic, 1 indicates that the quality ranking sphere region is on the left view of stereoscopic content, 2 indicates that the quality ranking sphere region is on the right view of stereoscopic content, 3 indicates that the quality ranking sphere region is on both the left and right views.

`quality_ranking` specifies a quality ranking value of the quality ranking sphere region. `quality_ranking` equal to 0 indicates that the quality ranking value is not defined. The semantics of non-zero quality ranking values are specified in subclause 7.8.1.

`view_idc` equal to 0 indicates that the quality ranking sphere region is monoscopic, 1 indicates that the quality ranking sphere region is on the left view of stereoscopic content, 2 indicates that the quality ranking sphere region is on the right view of stereoscopic content, 3 indicates that the quality ranking sphere region is on both the left and right views. When not present, the value of `view_idc` is inferred to be equal to the value of `default_view_idc`.

`orig_width` and `orig_height` specify the width and height, respectively, of such a monoscopic projected picture for which both horRatio and verRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking sphere region, are equal to 1.

NOTE 2 `orig_width` and `orig_height` represent the width and height of the picture from which the packed region has been extracted without resampling.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

NOTE 3 A player is recommended to parse SphereRegionQualityRankingBox and select the track for playing that matches the user's viewing orientation in a manner that:

- The quality ranking value on the region covering the viewport is greater than 0 and less than that for other regions.
- The resolution of the region covering the viewport is suitable for the display. If quality_type is equal to 1, orig_width and orig_height represent the width and height of the monoscopic projected picture from which the packed region covering the viewport has been extracted. Otherwise, width and height of VisualSampleEntry could be used to conclude the resolution on the viewport.

SphereRegionStruct(1) specifies the spherical location and size of the quality ranking sphere region relative to the global coordinate axes, while the shape type value of the quality ranking sphere regions is indicated by region_definition_type. The value of interpolate in SphereRegionStruct(1) shall be equal to 0.

7.8.3 2D region-wise quality ranking

7.8.3.1 Definition

Box type:	'2dqr'
Container:	VisualSampleEntry
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

7.8.3.2 Syntax

```
aligned(8) class 2DRegionQualityRankingBox extends FullBox('2dqr', 0, 0) {
      2DRegionQualityRankingStruct();
}

aligned(8) class 2DRegionQualityRankingStruct() {
   unsigned int(8) num_regions;
   unsigned int(1) remaining_area_flag;
   unsigned int(1) view_idc_presence_flag;
   unsigned int(1) quality_ranking_local_flag;
   unsigned int(4) quality_type;
   bit(1) reserved = 0;
   if (view_idc_presence_flag == 0) {
      unsigned int(2) default_view_idc;
      bit(6) reserved = 0;
   }
   for (i = 0; i < num_regions; i++) {
      unsigned int(8) quality_ranking;
      if (view_idc_presence_flag == 1) {
         unsigned int(2) view_idc;
         bit(6) reserved = 0;
      }
      if (quality_type == 1) {
         unsigned int(16) orig_width;
         unsigned int(16) orig_height;
      }
      if ((i < (num_regions - 1)) || (remaining_area_flag == 0)) {
         unsigned int(16) left_offset;
         unsigned int(16) top_offset;
         unsigned int(16) region_width;
         unsigned int(16) region_height;
      }
   }
}
```

7.8.3.3 Semantics

quality_ranking, view_idc_presence_flag, default_view_idc, and view_idc are specified identically to the syntax elements with the same names in SphereRegionQualityRankingBox but are further constrained as follows:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

– When a 2DRegionQualityRankingBox is present in a VisualSampleEntry and StereoVideoBox is present in the VisualSampleEntry and indicates temporal interleaving, view_idc_presence_flag shall be equal to 0 and default_view_idc shall be equal to 3.

num_regions specifies the number of quality ranking 2D regions for which the quality ranking information is given in this box. Value 0 is reserved. There shall be no pixel of the decoded picture that is contained in more than one of these quality ranking 2D regions.

remaining_area_flag equal to 0 specifies that all the quality ranking 2D regions are defined by the left_offset, top_offset, region_width, and region_height. remaining_area_flag equal to 1 specifies that the first num_regions − 1 quality ranking 2D regions are defined by left_offset, top_offset, region_width, and region_height and the last remaining quality ranking 2D region is the area in the picture with width equal to width of VisualSampleEntry and height equal to height of VisualSampleEntry, not covered by the union of the first num_regions − 1 quality ranking 2D regions. The last remaining quality ranking 2D region may be on both the left and right views.

quality_ranking_local_flag equal to 1 specifies that the quality ranking information provided in this instance of this box is relative to the quality ranking information provided in all instances of this box for this track only. quality_ranking_local_flag equal to 0 specifies that the quality ranking information provided in this instance of this box is relative to the quality ranking information provided in all instances of SphereRegionQualityRankingBox and 2DRegionQualityRankingBox with quality_ranking_local_flag equal to 0 for this and other tracks.

NOTE 1 quality_ranking_local_flag equal to 1 could be used in a track that includes coded video data by reference to a set of tracks among which one is selected as the source of the coded video data. For example, an 'alte' track reference specified in subclause 7.2.1 could be used for this purpose. SphereRegionQualityRankingBox or 2DQualityRankingBox with quality_ranking_local_flag equal to 0 could be present in the referenced tracks within the file.

quality_type indicates which factor causes the differences in the quality of packed regions on the picture. quality_type equal to 0 specifies that all packed regions correspond to the same projected picture resolution. quality_type equal to 1 specifies that at least one horRatio value, as derived in subclause 5.4.2, may differ from other horRatio values among all pairs of packed and projected regions of the picture or at least one verRatio value, as derived in subclause 5.4.2, may differ from other verRatio values among all pairs of packed and projected regions of the picture. quality_type values greater than 1 are reserved.

orig_width and orig_height specify the width and height, respectively, of such a monoscopic projected picture for which both horRatio and verRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking 2D region, are equal to 1.

NOTE 2 orig_width and orig_height represent the width and height of the picture from which the packed region has been extracted without resampling.

NOTE 3 A player is recommended to parse 2DRegionQualityRankingBox and select the track for playing that matches the user's viewing orientation in a manner that:

- The quality ranking value on the region covering the viewport is greater than 0 and less than that for other regions.
- The resolution of the region covering the viewport is suitable for the display. If quality_type is equal to 1, orig_width and orig_height represent the width and height of the monoscopic projected picture from which the packed region covering the viewport has been extracted. Otherwise, width and height of VisualSampleEntry could be used to conclude the resolution on the viewport.

left_offset, top_offset, region_width, and region_height are integer values that indicate the position and size of the quality ranking 2D region. left_offset and top_offset indicate the horizontal and vertical coordinates, respectively, of the upper left corner of the quality ranking 2D region within the picture, in units of luma samples. region_width and region_height indicate the width and height, respectively, of the quality ranking 2D region within the picture, in units of luma samples. left_offset + region_width shall be less than width of VisualSampleEntry. top_offset + region_height shall be less than height of VisualSampleEntry.

region_width shall be greater than 0.

region_height shall be greater than 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

7.9 Storage of omnidirectional images

7.9.1 General

Omnidirectional images are stored in a file as image items, as specified in ISO/IEC 23008-12. The `ProjectionFormatProperty` shall be present for an omnidirectional image item. When an image item contains stereoscopic content, the `FramePackingProperty` shall be present for the image item. When an image item contains a packed picture generated by region-wise packing from the projected picture, the `RegionWisePackingProperty` shall be present for the image item.

7.9.2 Frame packing item property

7.9.2.1 Definition

Box type:	`'stvi'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`FramePackingProperty` indicates that the reconstructed image contains a representation of two spatially packed constituent pictures.

`essential` shall be equal to 1 for a `'stvi'` item property.

7.9.2.2 Syntax

`FramePackingProperty` has the same syntax as `StereoVideoBox` specified in ISO/IEC 14496-12.

7.9.2.3 Semantics

The semantics of the syntax elements within the `FramePackingProperty` are the same as those specified for the syntax elements of `StereoVideoBox` as defined in ISO/IEC 14496-12 and in subclause 7.1.10.

7.9.3 Projection format item property

7.9.3.1 Definition

Box type:	`'prfr'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`ProjectionFormatProperty` indicates the omnidirectional projection format of the image.

When `'prfr'` is present, the reconstructed image represents a packed picture that has been generated as indicated in Figure 2 and Figure 3 for a monoscopic and stereoscopic image, respectively. The semantics of the sample locations of the reconstructed image are specified in subclause 7.5.1.2.

The format of the projected monoscopic pictures is indicated with the `ProjectionFormatProperty`.

For stereoscopic video, the frame packing arrangement of the projected left and right pictures is indicated with the `FramePackingProperty`. The absence of `FramePackingProperty` indicates that the content of the image item is monoscopic.

When both `ProjectionFormatProperty` and `FramePackingProperty` are present for an image item, `stereo_scheme` shall be equal to 4, and the first byte of `stereo_indication_type` shall be equal to 3 or 4, and the second byte of `stereo_indication_type` shall be equal to 0 indicating that quincunx sampling is not in use.

Optional region-wise packing is indicated with the `RegionWisePackingProperty`. The absence of `RegionWisePackingProperty` indicates that no region-wise packing is applied.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`essential` shall be equal to 1 for a `'prfr'` item property.

The values of the variables HorDiv1 and VerDiv1 are set as follows for an image item:

- If `FramePackingProperty` is not present for the image item, HorDiv1 is set equal to 1 and VerDiv1 is set equal to 1.
- Otherwise (`FramePackingProperty` is present for the image item), the following applies:
  - If side-by-side frame packing is indicated, HorDiv1 is set equal to 2 and VerDiv1 is set equal to 1.
  - Otherwise (top-bottom frame packing is indicated), HorDiv1 is set equal to 1 and VerDiv1 is set equal to 2.

The width and height of a monoscopic projected luma picture (ConstituentPicWidth and ConstituentPicHeight, respectively) corresponding to the reconstructed image of the image item are derived as follows:

- If `RegionWisePackingProperty` is not present for the image item, ConstituentPicWidth and ConstituentPicHeight are set to be equal to `image_width` / HorDiv1 and `image_height` / VerDiv1, respectively, where `image_width` and `image_height` are syntax elements of `ImageSpatialExtentsProperty` associated with the image item.
- Otherwise, ConstituentPicWidth and ConstituentPicHeight are set equal to `proj_picture_width` / HorDiv1 and `proj_picture_height` / VerDiv1, respectively.

If `RotationProperty` is not present for the image item, RotationFlag is set equal to 0. Otherwise, RotationFlag is set equal to 1.

If `FramePackingProperty` is not present for the image item, SpatiallyPackedStereoFlag, TopBottomFlag, and SideBySideFlag are set equal to 0. Otherwise, the following applies:

- When the `FramePackingProperty` indicates top-bottom frame packing, SpatiallyPackedStereoFlag is set equal to 1, TopBottomFlag is set equal to 1, and SideBySideFlag is set equal to 0.
- When the `FramePackingProperty` indicates side-by-side frame packing, SpatiallyPackedStereoFlag is set equal to 1, TopBottomFlag is set equal to 0, and SideBySideFlag is set equal to 1.

If `RegionWisePackingProperty` is not present for the image item, RegionWisePackingFlag is set equal to 0. Otherwise, RegionWisePackingFlag is set equal to 1.

7.9.3.2 Syntax

```
aligned(8) class ProjectionFormatProperty
extends ItemFullProperty('prfr', 0, 0) {
   ProjectionFormatStruct(); /* specified in clause 7.5.2 */
}
```

7.9.4 Essential fisheye image item property

7.9.4.1 General

When an image item contains a picture which consists of multiple circular images captured by fisheye cameras, the `EssentialFisheyeImageProperty` shall be present for the image item.

7.9.4.2 Definition

Box type:	`'fovi'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`EssentialFisheyeImageProperty` provides essential fisheye parameters for stitching and rendering fisheye images at the OMAF player. The fields in `EssentialFisheyeImageProperty` provide region information of circular images in the image item and the field of view and camera parameters of the fisheye lens(es).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

When the 'fovi' item property is present, the 'prfr' item property shall not be present.

essential shall be equal to 1 for a 'fovi' item property.

7.9.4.3 Syntax

```
aligned(8) class EssentialFisheyeImageProperty
extends ItemFullProperty('fovi', 0, 0) {
   FisheyeVideoEssentialInfoStruct(); /*specified in clause 6*/
}
```

7.9.4.4 Semantics

The semantics are as specified in subclause 6.2.2.

7.9.5 Supplemental fisheye image item property

7.9.5.1 Definition

Box type:	'fvsi'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

SupplementalFisheyeImageProperty provides supplemental fisheye parameters, such as the local field of view information for high quality stitching and rendering as well as deadzone information.

When the 'fvsi' item property is present, the 'fovi' item property shall be present.

7.9.5.2 Syntax

```
aligned(8) class SupplementalFisheyeImageProperty
extends ItemFullProperty('fvsi', 0, 0) {
   FisheyeVideoSupplementalInfoStruct(); /*specified in clause 6*/
}
```

7.9.5.3 Semantics

The semantics are as specified in subclause 6.3.2.

7.9.6 Region-wise packing item property

7.9.6.1 Definition

Box type:	'rwpk'
Property type:	Descriptive item property
Container:	ItemPropertyContainerBox
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

RegionWisePackingProperty specifies the mapping between packed regions and the corresponding projected regions and specifies the location and size of the guard bands, if any.

essential shall be equal to 1 for a 'rwpk' item property.

7.9.6.2 Syntax

```
aligned(8) class RegionWisePackingProperty
extends ItemFullProperty('rwpk', 0, 0) {
   RegionWisePackingStruct(); /* specified in clause 7.5.3 */
}
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

7.9.6.3 Semantics

Subclause 7.5.3 applies with the following additional constraint:

- `packed_picture_width` and `packed_picture_height` shall have such values that `packed_picture_width` is an integer multiple of `image_width` and `packed_picture_height` is an integer multiple of `image_height`, where `image_width` and `image_height` are syntax elements of the `ImageSpatialExtentsProperty` associated to the image item.

7.9.7 Rotation item property

7.9.7.1 General

When an image item contains an omnidirectional image specified based on the local coordinate axes, the `RotationProperty` shall be present for the image item.

7.9.7.2 Definition

Box type:	`'rotn'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`RotationProperty` is used to indicate the yaw, pitch, and roll angles, respectively, of the rotation to be applied to convert the local coordinate axes to the global coordinate axes. In the case of stereoscopic omnidirectional image, the fields apply to each view individually. The absence of `RotationProperty` indicates that `rotation_yaw`, `rotation_pitch`, and `rotation_roll` are all inferred to be equal to 0.

`essential` shall be equal to 1 for a `'rotn'` item property.

7.9.7.3 Syntax

```
aligned(8) class RotationProperty
extends ItemFullProperty('rotn', 0, 0) {
   RotationStruct(); /* specified in clause 7.5.4 */
}
```

7.9.8 Coverage information item property

7.9.8.1 General

7.9.8.2 Definition

Box type:	`'covi'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`CoverageInformationProperty` is used to indicate the content coverage of the omnidirectional image.

NOTE It is totally up to the OMAF player to handle the area that is not covered by the content when rendering the omnidirectional image content.

Each sphere location within the sphere regions specifying the content coverage shall have a corresponding sample in the reconstructed image. However, there may be some sphere locations that do have corresponding samples in the reconstructed image but are outside the content coverage.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

7.9.8.3 Syntax

```
aligned(8) class CoverageInformationProperty
extends ItemFullProperty('covi', 0, 0) {
   ContentCoverageStruct()
}
```

7.9.9 Initial viewing orientation item property

7.9.9.1 Definition

Box type:	`'iivo'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per an item):	No
Quantity (per an item):	Zero or one

`InitialViewingOrientationProperty` indicates the initial viewing orientation according to which the image should be initially rendered to the user. In the absence of this property `centre_azimuth`, `centre_elevation`, and `centre_tilt` should all be inferred to be equal to 0.

An OMAF player should use the indicated or inferred `centre_azimuth`, `centre_elevation`, and `centre_tilt` values as follows:

- If the orientation/viewport metadata of the OMAF player is obtained on the basis of an orientation sensor included in or attached to a viewing device, the OMAF player should
  - obey only the `centre_azimuth` value, and
  - ignore the values of `centre_elevation` and `centre_tilt` and use the respective values from the orientation sensor instead.
- Otherwise, the OMAF player should obey all three of `centre_azimuth`, `centre_elevation`, and `centre_tilt`.

For the syntax and semantics of `InitialViewingOrientationProperty`, `num_regions` is inferred to be equal to 1, `shape_type` is inferred to be equal to 0, `dynamic_range_flag` is inferred to be equal to 0, `static_azimuth_range` is inferred to be equal to 0, and `static_elevation_range` is inferred to be equal to 0.

NOTE This metadata applies to any viewport regardless of which azimuth and elevation ranges are covered by the viewport. Thus, `dynamic_range_flag`, `static_azimuth_range`, and `static_elevation_range` do not affect the dimensions of the viewport that this metadata concerns and are hence required to be equal to 0. When the OMAF player obeys the `centre_tilt` value as concluded above, the `centre_tilt` value of could be interpreted by setting the azimuth and elevation ranges for the sphere region of the viewport equal to those that are actually used in displaying the viewport.

7.9.9.2 Syntax

```
aligned(8) class InitialViewingOrientationProperty
extends ItemFullProperty('iivo', 0, 0) {
   InitialViewingOrientationSample vr_initial_orientation;
}
```

7.9.9.3 Semantics

The semantics of the `InitialViewingOrientationSample` structure are specified in subclause 7.7.4.3 and apply to the image item by replacing the phrase "sample" with the phrase "image item" except that the semantics of `refresh_flag` are undefined and the flag shall be equal to 0.

7.10 Storage of timed text for omnidirectional video

7.10.1 General

Timed text is used for providing subtitles and closed captions for omnidirectional video. Timed text cues may be rendered on a certain region relative to the sphere (i.e., only visible when the user looks in a specific direction), or it may be rendered

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

in a region on the current viewport (i.e., always visible irrespective of the viewing direction), in which case the text/cue region positions are relative to the current viewport.

For the timed text of the omnidirectional video, W3C Recommendation, *TTML profiles for Internet media subtitles and captions 1.0 (IMSC1)* or WebVTT: *The web video text tracks format, W3C* shall be used.

The IMSC1 or WebVTT streams shall be carried in tracks conforming to ISO/IEC 14496-30.

`XMLSubtitleSampleEntry` or `WVTTSampleEntry` shall include the `OmafTimedTextConfigBox` as specified in subclause 7.10.2.

7.10.2 OMAF timed text configuration box

7.10.2.1 Definition

Box Type: `'otcf'`
Container: `XMLSubtitleSampleEntry` or `WVTTSampleEntry`
Mandatory: Yes (for timed text tracks associated with an omnidirectional video track)
Quantity: One (for timed text tracks associated with an omnidirectional video track)

This box provides configuration information for presenting timed text together with omnidirectional video.

7.10.2.2 Syntax

```
class OmafTimedTextConfigBox extends FullBox('otcf', 0, 0) {
   unsigned int(1) relative_to_viewport_flag;
   unsigned int(1) relative_disparity_flag;
   unsigned int(1) depth_included_flag;
   bit(5) reserved = 0;
   unsigned int(8) region_count;
   for (i=0;i<region_count;i++) {
      string region_id;
      if(relative_to_viewport_flag == 1) {
         if (relative_disparity_flag)
            signed int(16) disparity_in_percent;
         else
            signed int(16) disparity_in_pixels;
      } else {
         SphereRegionStruct(0);
         if (depth_included_flag)
            unsigned int(16) region_depth;
      }
   }
}
```

7.10.2.3 Semantics

`relative_to_viewport_flag` specifies how the timed text cues are to be rendered. The value 1 indicates that the timed text is expected to be always present on the display screen, i.e., the text cue is visible independently of the viewing direction of the user. The value 0 indicates that the timed text is expected to be rendered at a certain position on the sphere, i.e., the text cue is only visible when the user is looking in the direction where the text cue is rendered.

NOTE 1 When `relative_to_viewport_flag` is equal to 1, the active area where the timed text could be displayed is provided by the timed text track as a rectangular region.

`relative_disparity_flag` indicates whether the disparity is provided as a percentage value of the width of the display window for one view (when the value is equal to 1) or as a number of pixels (when the value is equal to 0).

`depth_included_flag` equal to 1 indicates that the depth (z-value) of regions on which the timed text is to be rendered is present. The value 0 indicates that the depth (z-value) of regions on which the timed text is to be rendered is not present.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

`region_count` specifies the number of text regions for which a placement inside the sphere is provided. Each region is identified by an identifier. (both WebVTT and TTML identify regions using a unique id). When a timed metadata track containing the timed text sphere metadata track is present and linked to this timed text track by the track reference of type `'cdsc'`, the value of `region_count` shall be 0.

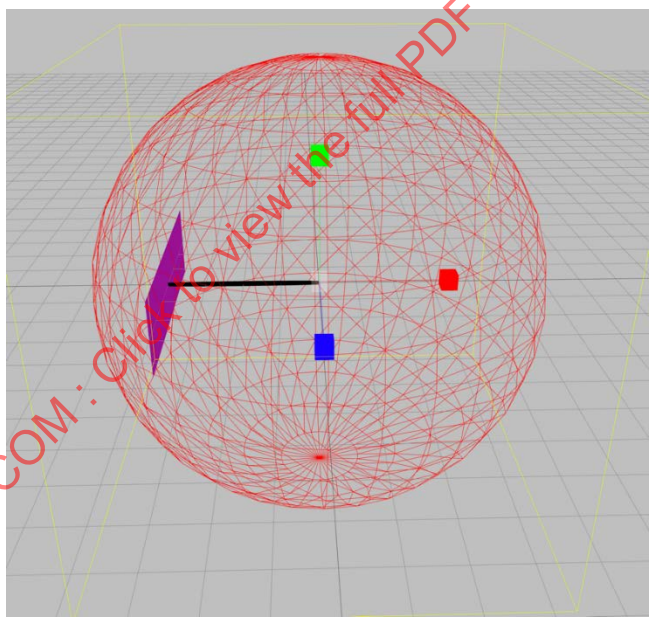
NOTE 2 Both WebVTT and TTML identify a region using a unique identifier.

`region_id` provides the identifier of the text region. This identifier shall be equal to the identifier of the corresponding region defined in timed text streams in the IMSC1 or WebVTT track.

`disparity_in_percent` indicates the disparity, in units of 2^{-16}, as a fraction of the width of the display window for one view. The value may be negative, in which case the displacement direction is reversed. This value is used to displace the region to the left on the left eye view and to the right on the right eye view.

`disparity_in_pixels` indicates the disparity in pixels. The value may be negative, in which case the displacement direction is reversed. This value is used to displace the region to the left on the left eye view and to the right on the right eye view.

`SphereRegionStruct()` indicates a sphere location that is used, together with other information, to determine where the timed text is placed and displayed in 3D space. The vector between the centre of the sphere and this sphere location is the normal vector to the rendering 3D plane on which the timed text cue is to be rendered. This information and the depth of the 3D plane are used to determine the position of the rendering 3D plane in 3D space on which the timed text cue is to be rendered. An example of such region is shown in Figure 21.



NOTE The purple plane indicates the region, and the line segment in black indicates the depth.

Figure 21 – An example of a region on which the timed text is rendered

When `SphereRegionStruct()` is included in the `OmafTimedTextConfigBox`, the following applies:

For the syntax and semantics of `SphereRegionStruct()` included in the `OmafTimedTextConfigBox`, the values of `shape_type`, `dynamic_range_flag`, `static_azimuth_range`, and `static_elevation_range` are all inferred to be equal to 0.

`centre_azimuth` and `centre_elevation` specify the sphere location that is used, together with other information, to determine where the timed text is placed and displayed in 3D space. `centre_azimuth` shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. `centre_elevation` shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive.

`centre_tilt` shall be equal to 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`region_depth` indicates the depth (z-value) of the region on which the timed text is to be rendered. The depth value is the norm of the normal vector of the timed text region. This value is relative to a unit sphere and is in units of 2^{-16}.

7.10.2.4 Rendering procedure (informative)

The procedure of this clause specifies how the timed text cues are suggested to be rendered.

During rendering, some adjustment to the styles of the timed text could be desirable or necessary, to achieve better visibility of the timed text cues based on depth of the text and user preferences.

The rendering procedure for timed text on omnidirectional video depends on the value of `relative_to_viewport_flag`.

If `relative_to_viewport_flag` is equal to 0, the rendering procedure may be as follows:

1. Setup the VR scene geometry by creating the sphere and placing the rendering camera in the centre of the sphere (depending on whether the content is stereo or mono, the rendering camera has to correspondingly be mono or stereo).
2. For each text sample to be rendered at time t, the following applies:
   a. Fetch the corresponding information about depth d, direction (u, v), and the 2D dimensions of the region.
   b. Create a 3D plane with the normal vector calculated out of (d, u, v).
   c. Render the text cue on that plane and on the corresponding 2D rectangle with a centre at the normal vector.

Otherwise (`relative_to_viewport_flag` is equal to 1), the rendering procedure may be as follows:

1. Read information of the rectangular region from the timed text track and the disparity disparityVal for each text cue at time t.
2. After rendering the omnidirectional video, render the text cue in the rectangular region for the left or right half of the screen as follows:
   a. For the left half of the screen, render the text cue in the left-half of the rectangular region such that the left-side boundary of the text cue starts from the position that is disparityVal units to the right from the left-side boundary of the left-side half of the rectangular region.
   b. For the right half of the screen, render the text cue in the right-half of the rectangular region such that the left-side boundary of the text cue starts from the position that is disparityVal units to the right from the left-side boundary of the right-side half of the rectangular region.

7.10.3 IMSC1 tracks

When IMSC1 streams are used, `OmafTimedTextConfigBox` shall be present in `XMLSubtitleSampleEntry`. When the disparity information is present in `OmafTimedTextConfigBox`, it is superseded by the `tts:disparity` attribute of the IMSC1 track.

```
class XMLSubtitleSampleEntry() extends SubtitleSampleEntry ('stpp') {
   string    namespace;
   string    schema_location;  // optional
   string    auxiliary_mime_types;
                    // optional, required if auxiliary resources are present
   OmafTimedTextConfigBox();         // optional
}
```

The `namespace` field of the `XMLSubtitleSampleEntry` shall contain one instance of the string `"http://www.w3.org/ns/ttml"`.

The `schema_location` field of the `XMLSubtitleSampleEntry` shall contain one of following string instances:

- `"http://www.w3.org/ns/ttml/profile/imsc1/text"`
- `"http://www.w3.org/ns/ttml/profile/imsc1/image"`

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

The `XMLSubtitleSampleEntry` shall contain a `MIMEBox` and its `content_type` field shall be constrained as follows:

- The `type` shall be `"application"`.
- The `subtype` shall be `"ttml+xml"`.
- The `codecs` parameter shall contain either `"im1t"` or `"im1i"` that indicates that an IMSC1 Text or Image processor is required, respectively.

All samples of the IMSC1 track shall conform to the Text Profile or Image Profile specified in W3C Recommendation, *TTML profiles for Internet media subtitles and captions 1.0*.

All IMSC1 sample format shall conform to ISO/IEC 14496-30.

The media type of the IMSC1 samples is `"application/mp4"`.

7.10.4 WebVTT tracks

When WebVTT is used, `OmafTimedTextConfigBox` shall be present in `WVTTSampleEntry`.

```
class WVTTSampleEntry() extends PlainTextSampleEntry ('wvtt') {
   WebVTTConfigurationBox  config;
   WebVTTSourceLabelBox    label;    // recommended
   OmafTimedTextConfigBox();         // optional
}
```

All WebVTT samples shall conform to the sample format as defined in ISO/IEC 14496-30:2018 subclause 6.6.

The media type of the WebVTT samples is `"text/vtt"`.

8 Omnidirectional media encapsulation and signalling in DASH

8.1 Architecture of DASH delivery in OMAF

Figure 22 illustrates the content flow in the DASH delivery function of OMAF.

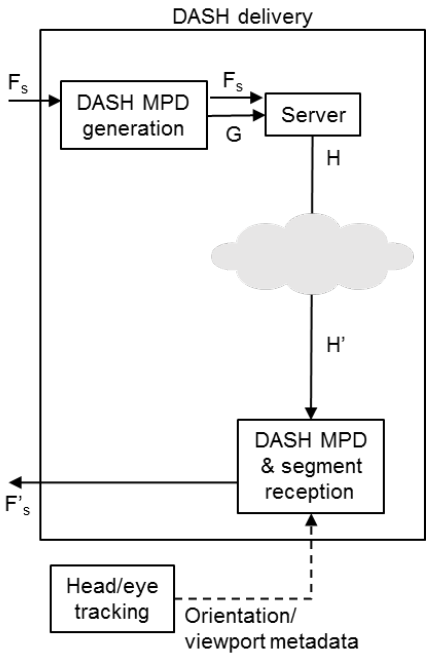


Figure 22 – Content flow in the DASH delivery function of OMAF

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

The following interfaces are normatively specified in this clause:

– F_s/F'_s: initialization and media segments; as defined generally below and specified for media profiles in Annex B;

– G: DASH media presentation description (MPD), including omnidirectional media-specific metadata, such as information on projection and region-wise packing, as specified in subclause 8.2.

An MPD (G) is generated based on the segments (F_s) and other media files representing the same content. The DASH MPD generator includes omnidirectional media-specific descriptors as specified in subclause 8.2. The descriptors include projection type, region-wise packing type, content coverage, spherical region-wise quality ranking, 2D region-wise quality ranking, and fisheye omnidirectional video information. These information may be generated on the basis of the equivalent information in the segments.

A DASH client obtains a current viewing orientation or viewport e.g. from the head-mounted display that detects the head and possibly also eye orientation. By parsing metadata, e.g. on projection and region-wise packing, from the MPD, the DASH client concludes which Adaptation Set and Representation contain the supported projection format etc. and which Adaptation Set and Representation cover the current viewing orientation at the highest quality and at a bitrate that may be afforded by the prevailing estimated network throughput. The DASH client issues (sub)segment requests accordingly.

The server typically provides segments (F_s) to the client. The server may also provide the MPD (considered as part of interface H in this case), or the MPD may be delivered by other means to the client. The segments and MPD are delivered over a network, and the received segments and MPD from the server are marked with H' in the figure. The output from the server (H) is considered to be identical to the input to the DASH MPD and segment reception block (H'). The received segments (F'_s) are output by the DASH MPD and segment reception block to the file/segment decapsulation block (see subclauses 4.2 and 4.3). In some media profiles, the segment reception function may include reconstruction of a conforming segment sequence, which is regarded as the F'_s interface.

8.2 Usage of DASH in OMAF

8.2.1 General

The following applies for the specifications in subclauses 8.2 and 8.3:

– The presence of an element at MPD level refers to that the element is a child element of the **`MPD`** element.

– The presence of an element at Adaptation Set level refers to that the element is a child element of an **`AdaptationSet`** element.

– The presence of an element at Representation level refers to that the element is a child element of a **`Representation`** element.

8.2.2 Signalling of stereoscopic frame packing

A DASH **`FramePacking`** element with a `@schemeIdUri` attribute equal to `urn:mpeg:mpegB:cicp:VideoFramePackingType` may be present at Adaptation Set level and shall not be present at MPD or Representation level. When used with projected omnidirectional video (i.e., when the PF descriptor is present), this element indicates that the projected pictures consist of spatially or temporally packed constituent pictures of the left and right views. The `@value` of the **`FramePacking`** element specifies the frame packing type for the stereoscopic video. This value shall be equal to 3, 4, or 5 with the meaning of those values as specified for `VideoFramePackingType` in ISO/IEC 23091-2. The value of `QuincunxSamplingFlag` as specified in ISO/IEC 23091-2 is inferred to be equal to 0.

8.2.3 Carriage of timed metadata

A timed metadata track, e.g., of track sample entry type `'invo'` or `'rcvp'` as specified in subclause 7.7, may be encapsulated in a DASH Representation. The `@associationId` attribute of this metadata Representation shall contain one or more values of the `@id` attribute of the Representation(s) containing the omnidirectional media carried by the media track(s) that are associated with the timed metadata track through a `'cdsc'` track reference as specified in subclause 7.1.5.1. The `@associationType` attribute of this metadata Representation shall be equal to `'cdsc'`.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

8.3 DASH MPD descriptors for omnidirectional media

8.3.1 XML namespace and schema

A number of new XML elements and attributes are defined and used. These new XML elements are defined in a separate namespace "urn:mpeg:mpegI:omaf:2017". The namespace designator "omaf:" is used to refer to this name space in this document. These are defined in normative schema documents in each clause where a new MPD descriptor is specified. The namespace designator "xs:" shall correspond to namespace http://www.w3.org/2001/XMLSchema as defined in W3C Recommendation, *XML schema part 1: Structures*. Items in the "Data type" column of tables in clause 8.2 use datatypes defined in XML Schema Part 2 and shall have the meaning as defined in W3C Recommendation, *XML schema part 2: Datatypes*.

8.3.2 Signalling of projection type information

An **EssentialProperty** element with a @schemeIdUri attribute equal to "urn:mpeg:mpegI:omaf:2017:pf" is referred to as a projection format (PF) descriptor.

At most one PF descriptor may be present at MPD level. At most one PF descriptor may be present at Adaptation Set level. At most one PF descriptor may be present at Representation level.

The omaf:@projection_type attribute of a PF descriptor present at a hierarchically lower level overrides omaf:@projection_type attribute of a PF descriptor present at a hierarchically higher level. For example, when both an **AdaptationSet** element and a **Representation** element in the **AdaptationSet** element have a PF descriptor present, the PF descriptor present in the **Representation** element applies to the Representation.

The @value attribute of the PF descriptor shall not be present. The PF descriptor shall include an omaf:@projection_type attribute whose value shall not be empty as specified in Table 12.

Table 12 – Semantics of omaf:@projection_type attribute

Attribute for PF descriptor	Use	Data type	Description
omaf:@projection_type	M	omaf:listofUnsignedByte	Specifies a list of projection type values of the projected picture as specified in Table 8. Each value in the list shall be in the range of 0 to 31, inclusive. The values 32 to 255 are reserved. Each value in the list shall be unique. For ISO base media file format Segments, projection_type shall be equal to projection_type in ProjectionFormatBox in sample entries of the Initialization Segment.

The data type for the attribute shall be as defined in the XML schema. An XML schema for projection type signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:omaf:2017 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">
    <xs:attribute name="projection_type" type="omaf:listOfUnsignedByte"/>
    <xs:simpleType name="listOfUnsignedByte">
     <xs:restriction>
         <xs:simpleType>
             <xs:list itemType="xs:unsignedByte"/>
         </xs:simpleType>
         <xs:minLength value="1"/>
     </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

8.3.3 Signalling of region-wise packing type

An **EssentialProperty** element with a @schemeIdUri attribute equal to "urn:mpeg:mpegI:omaf:2017:rwpk" is referred to as a region-wise packing (RWPK) descriptor.

An RWPK descriptor indicates the applied region-wise packing types in the Representation(s) associated with the descriptor.

At most one RWPK descriptor may be present at MPD level. At most one RWPK descriptor may be present at Adaptation Set level. At most one RWPK descriptor may be present at Representation level.

The omaf:@packing_type attribute of a RWPK descriptor present at a hierarchically lower level overrides omaf:@packing_type attribute of a RWPK descriptor present at a hierarchically higher level. For example, when both an **AdaptationSet** element and a **Representation** element in the **AdaptationSet** element have a RWPK descriptor present, the RWPK descriptor present in the **Representation** element applies to the Representation.

The @value of the RWPK descriptor shall not be present. The RWPK descriptor may include an omaf:@packing_type attribute as specified in Table 13.

Table 13 – Semantics of omaf:@packing_type attribute

Attribute for RWPK descriptor	Use	Data type	Description
omaf:@packing_type	O	omaf:OptionallistofUnsignedByte	Specifies a list of the packing type value of the picture as specified in Table 9. Each value in the list shall be in the range of 0 to 15, inclusive. The values 16 to 255 are reserved. Each value in the list shall be unique. For ISO base media file format Segments, packing_type shall be equal to packing_type in RegionWisePackingBox in sample entries of the Initialization Segment. When this omaf:@packing_type attribute is not present or does not include any value in a RWPK descriptor, omaf:@packing_type value is inferred to be equal to 0.

The absence of a RWPK descriptor indicates that no region-wise packing has been applied.

The data type for the attribute shall be as defined in the XML schema. An XML schema for region-wise packing type signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:omaf:2017 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">
    <xs:attribute name="packing_type" type="omaf:OptionallistOfUnsignedByte"/>
    <xs:simpleType name="OptionallistOfUnsignedByte">
     <xs:restriction>
         <xs:simpleType>
             <xs:list itemType="xs:unsignedByte"/>
         </xs:simpleType>
         <xs:minLength value="0"/>
     </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

8.3.4 Signalling of content coverage

A **SupplementalProperty** element with a @schemeIdUri attribute equal to "urn:mpeg:mpegI:omaf:2017:cc" is referred to as a content coverage (CC) descriptor.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

At most one CC descriptor may be present at Adaptation Set level. A CC descriptor shall not be present at MPD or Representation level.

The CC descriptor indicates that each Representation covers the sphere region as specified in subclause 7.7 by `shape_type` and syntax elements `centre_azimuth`, `centre_elevation`, `centre_tilt`, `azimuth_range`, and `elevation_range` in `SphereRegionStruct` as included in the CC descriptor.

When the CC descriptor that applies to a Representation is present and the `CoverageInformationBox` is present in the track corresponding to the Representation, the CC descriptor shall carry equivalent information as the `CoverageInformationBox`.

The `@value` attribute of the CC descriptor shall not be present. The CC descriptor shall include elements and attributes as specified in Table 14.

Table 14 – Semantics of elements and attributes of CC descriptor

Elements and attributes for CC descriptor	Use	Data type	Description
cc	0..1	omaf:CCType	Container element whose attributes and elements specify sphere region coverage information.
cc`@shape_type`	O	xs:unsignedByte	Specifies the shape type of the sphere region, as specified in subclause 7.7.2.3. When not present, **cc**`@shape_type` is inferred to be equal to 0.
cc`@view_idc_presence_flag`	O	xs:boolean	Value 0 specifies that **cc.coverageInfo**`@view_idc` is not signalled. Value 1 specifies that **cc.coverageInfo**`@view_idc` is signalled and indicates the association of sphere regions with particular (left, right, or both) views or monoscopic content. When not present, **cc**`@vview_idc_presence_flag` is inferred to be equal to 0.
cc`@default_view_idc`	CM	omaf:ViewType	Value 0 indicates that all the sphere regions are monoscopic. Value 1 indicates that all the sphere regions are on the left view of a stereoscopic content. Value 2 indicates that all the sphere regions are on the right view of a stereoscopic content. Value 3 indicates that all the sphere regions are on both the left and right views. **cc**`@default_view_idc` shall be present when **cc**`@view_idc_presence_flag` is equal to 0. **cc**`@default_view_idc` shall be absent when **cc**`@view_idc_presence_flag` is equal to 1.
cc.coverageInfo	1..255	omaf:coverageInfoType	Element whose attribute **cc.coverageInfo**`@view_idc`, when present, provides information about view(s) to which coverage specified by sphere region defined by attributes **cc.coverageInfo**`@centre_azimuth`, **cc.coverageInfo**`@centre_elevation`, **cc.coverageInfo**`@centre_tilt`, **cc.coverageInfo**`@azimuth_range`, **cc.coverageInfo**`@elevation_range` applies.
cc.coverageInfo`@view_idc`	CM	omaf:ViewType	Value 1 indicates that the sphere region is on the left view of a stereoscopic content, value 2 indicates the sphere region is on the right view of a stereoscopic content, and value 3 indicates that the sphere region is on both the left and right views. Value 0 is reserved. **cc.coverageInfo**`@view_idc` shall be absent when **cc**`@view_idc_presence_flag` is equal to 0. **cc.coverageInfo**`@view_idc` shall be present when **cc**`@view_idc_presence_flag` is equal to 1.
cc.coverageInfo`@centre_azimuth`	O	omaf:Range1	Specifies the azimuth of the centre point of the sphere region in units of 2^{-16} degrees relative to the global coordinate axes. When not present, **cc.coverageInfo**`@centre_azimuth` is inferred to be equal to 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Elements and attributes for CC descriptor	Use	Data type	Description
cc.coverageInfo@centre_elevation	O	omaf:Range2	Specifies the elevation of the centre point of the sphere region in units of 2^{-16} degrees relative to the global coordinate axes. When not present, **cc.coverageInfo**@centre_elevation is inferred to be equal to 0.
cc.coverageInfo@centre_tilt	O	omaf:Range1	Specifies the tilt angle of the sphere region, in units of 2^{-16} degrees, relative to the global coordinate axes. When not present, **cc.coverageInfo**@centre_tilt is inferred to be equal to 0.
cc.coverageInfo@azimuth_range	O	omaf:HRange	Specifies the azimuth range of the sphere region through the centre point of the sphere region in units of 2^{-16} degrees. When not present **cc.coverageInfo**@azimuth_range is inferred to be equal to $360 * 2^{16}$.
cc.coverageInfo@elevation_range	O	omaf:VRange	Specifies the elevation range of the sphere region through the centre point of the sphere region in units of 2^{-16} degrees. When not present **cc.coverageInfo**@elevation_range is inferred to be equal to $180 * 2^{16}$.

The absence of the **cc** element in the CC descriptor indicates that each Representation covers the entire sphere when a PF descriptor that applies to the Representation is present.

When a PF descriptor is not present at MPD level or Adaptation Set level, there shall be no CC descriptor present in the **AdaptationSet** element.

The data types for various elements and attributes shall be as defined in the XML schema. An XML schema for the CC descriptor shall be as shown below. The schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:omaf:2017 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">
    <xs:element name="cc" type="omaf:CCType"/>
    <xs:complexType name="CCType">
       <xs:sequence>
            <xs:element name="coverageInfo" type="omaf:coverageInfoType"
            minOccurs="1" maxOccurs="255"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
            maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="shape_type" type="xs:unsignedByte" use="optional"
default="0"/>
     <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional"
default="0"/>
        <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="coverageInfoType">
        <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
        <xs:attribute name="centre_azimuth" type="omaf:Range1" use="optional"
default="0"/>
        <xs:attribute name="centre_elevation" type="omaf:Range2" use="optional"
default="0"/>
        <xs:attribute name="centre_tilt" type="omaf:Range1" use="optional"
default="0"/>
        <xs:attribute name="azimuth_range" type="omaf:HRange" use="optional"
default="23592960"/>
        <xs:attribute name="elevation_range" type="omaf:VRange" use="optional"
default="11796480"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
        <xs:simpleType name="Range1">
            <xs:restriction base="xs:int">
                <xs:minInclusive value="-11796480"/>
                <xs:maxInclusive value="11796479"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="Range2">
            <xs:restriction base="xs:int">
                <xs:minInclusive value="-5898240"/>
                <xs:maxInclusive value="5898240"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="HRange">
            <xs:restriction base="xs:unsignedInt">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="23592960"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="VRange">
            <xs:restriction base="xs:unsignedInt">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="11796480"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="ViewType">
            <xs:restriction base="xs:unsignedByte">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="3"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:schema>
```

8.3.5 Signalling of spherical region-wise quality ranking

A **SupplementalProperty** element with a `@schemeIdUri` attribute equal to `"urn:mpeg:mpegI:omaf:2017:srqr"` is referred to as a spherical region-wise quality ranking (SRQR) descriptor.

At most one SRQR descriptor for each **sphRegionQuality**`@shape_type` value of 0 and 1 may be present at Adaptation Set level. At most one SRQR descriptor for each **sphRegionQuality**`@shape_type` value of 0 and 1 may be present at Representation level. A SRQR descriptor shall not be present at MPD level.

The SRQR descriptor indicates a quality ranking value of a quality ranking sphere region relative to:

- other quality ranking sphere regions in the same Adaptation Set, and
- when **sphRegionQuality**`@quality_ranking_local_flag` is equal to 0, SRQR descriptors with **sphRegionQuality**`@quality_ranking_local_flag` equal to 0 and `@qualityRanking` values in all Adaptation Sets that have the same `@value` in the DASH **Viewpoint** element as the Adaptation Set containing this SRQR descriptor or containing the Representation that contains this SRQR descriptor.

NOTE 1 As specified in ISO/IEC 23009-1, the handling of the **Viewpoint** element is recommended to be applied equally for recognized and unrecognized `@schemeIdUri` values. It is suggested to use the same `@schemeIdUri` value for all **Viewpoint** elements in all Adaptation Sets of the same omnidirectional audio-visual content within the same Period.

The sphere region for the quality-ranking is specified by syntax elements `shape_type`, `centre_azimuth`, `centre_elevation`, `centre_tilt`, `azimuth_range`, `elevation_range` in `SphereRegionStruct` as specified in subclause 7.7. When the quality ranking value **sphRegionQuality.qualityInfo**`@quality_ranking` is non-zero, the picture quality within the entire indicated quality ranking sphere region is approximately constant.

When the SRQR descriptor that applies to a Representation is present and a `SphereRegionQualityRankingBox` with the same shape type as that in the SRQR descriptor is present in the track corresponding to the Representation, the SRQR descriptor shall carry equivalent information as the `SphereRegionQualityRankingBox`.

The `@value` attribute of the SRQR descriptor shall not be present. The SRQR descriptor shall include a **sphRegionQuality** element with its sub-elements and attributes as specified in Table 15.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Table 15 – Semantics of elements and attributes of SRQR descriptor

Elements and attributes for SRQR descriptor	Use	Data type	Description
sphRegionQuality	1	omaf:SphRegionQualityType	Container element which includes one or more quality information elements (**sphRegionQuality.qualityInfo**) and common set of attributes (**sphRegionQuality**@shape_type, **sphRegionQuality**@remaining_area_flag, **sphRegionQuality**@view_idc_presence_flag, **sphRegionQuality**@quality_ranking_local_flag, **sphRegionQuality**@quality_Type, **sphRegionQuality**@default_view_idc) that apply to all those quality information elements.
sphRegionQuality@shape_type	O	xs:unsignedByte	Value 0 specifies that the quality ranking sphere region is indicated through four great circles as specified in subclause 7.7.2.3. Value 1 specifies that the quality ranking sphere region is indicated through two azimuth and two elevation circles as specified in subclause 7.7.2.3. When not present **sphRegionQuality**@shape_type is inferred to be equal to 0.
sphRegionQuality@remaining_area_flag	O	xs:boolean	Value 0 specifies that all the quality ranking sphere regions are specified by the signalled **sphRegionQuality.qualityInfo** elements. Value 1 specifies that all except the last quality ranking sphere regions are specified by the signalled **sphRegionQuality.qualityInfo** elements, and the last remaining quality ranking sphere region is the sphere region within the content coverage, not covered by the union of the quality ranking sphere regions specified by the signalled **sphRegionQuality.qualityInfo** elements. When not present **sphRegionQuality**@remaining_area_flag is inferred to be equal to 0. The last remaining quality ranking sphere region may be on both the left and right views. NOTE 2 When **sphRegionQuality**@remaining_area_flag is equal to 1, the **qualityInfo** element is present for the last quality ranking sphere region but excludes @centre_azimuth, @centre_elevation, @centre_tilt, @azimuth_range, and @elevation_range attributes.
sphRegionQuality@view_idc_presence_flag	O	xs:boolean	Value 0 specifies that **sphRegionQuality.qualityInfo**@view_idc is not signalled in each **sphRegionQuality.qualityInfo** element. Value 1 specifies that **sphRegionQuality.qualityInfo**@view_idc is signalled and indicates the association of quality ranking sphere regions with particular (left or right or both) views or monoscopic content. When not present **sphRegionQuality**@view_idc_presence_flag is inferred to be equal to 0.
sphRegionQuality@quality_ranking_local_flag	O	xs:boolean	Value 0 specifies that the quality ranking information provided in this instance of this descriptor is relative to the quality ranking information provided in all instances of this descriptor with **sphRegionQuality**@quality_ranking_local_flag equal to 0 and in all instances of @qualityRanking in all Adaptation Sets that have the same @value in the DASH **Viewpoint** element as this Adaptation Set. Value 1 specifies that the quality ranking information provided in this instance of this descriptor is relative to the quality ranking information provided in all instances of this descriptor in this Adaptation Set only. When not present, the value of **sphRegionQuality**@quality_ranking_local_flag is inferred to be equal to 0. NOTE 3 **sphRegionQuality**@quality_ranking_local_flag equal to 1 might be used in the main Adaptation Set of a Preselection and **sphRegionQuality**@quality_ranking_local_flag equal to 0 could be used in Adaptation Sets that are not the main Adaptation Sets of a Preselection.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Elements and attributes for SRQR descriptor	Use	Data type	Description
sphRegionQuality@quality_type	M	omaf:QualityType	Indicates which factor causes the differences in the quality of packed regions on the picture. Value 0 specifies that all packed regions correspond to the same projected picture resolution. Value 1 specifies that at least one horRatio value, as derived in subclause 5.4.2, may differ from other horRatio values among all pairs of packed and projected regions of the picture or at least one verRatio value, as derived in subclause 5.4.2, may differ from other verRatio values among all pairs of packed and projected regions of the picture. Values greater 1 are reserved.
sphRegionQuality@default_view_idc	CM	omaf:ViewType	Value 0 indicates that all the quality ranking sphere regions are monoscopic. Value 1 indicates that all the quality ranking sphere regions are on the left view of stereoscopic content. Value 2 indicates that all the quality ranking sphere regions are on the right view of stereoscopic content. Value 3 indicates that all the quality ranking sphere regions are on both the left and right views. **sphRegionQuality**@default_view_idc shall be present when **sphRegionQuality**@view_idc_presence_flag is equal to 0. **sphRegionQuality**@default_view_idc shall be absent when **sphRegionQuality**@view_idc_presence_flag is equal to 1.
sphRegionQuality.qualityInfo	1..255	omaf:QualityInfoType	Element whose attribute **sphRegionQuality.qualityInfo**@quality_ranking provides quality ranking for one quality ranking sphere region described by its attributes **sphRegionQuality.qualityInfo**@view_idc, **sphRegionQuality.qualityInfo**@centre_azimuth, **sphRegionQuality.qualityInfo**@centre_elevation, **sphRegionQuality.qualityInfo**@centre_tilt, **sphRegionQuality.qualityInfo**@azimuth_range, **sphRegionQuality.qualityInfo**@elevation_range.
sphRegionQuality.qualityInfo@quality_ranking	M	xs:unsignedByte	Specifies a quality ranking value of the quality ranking sphere region. **sphRegionQuality.qualityInfo**@quality_ranking equal to 0 indicates that the quality ranking is not defined. When quality ranking sphere region A has a non-zero **sphRegionQuality.qualityInfo**@quality_ranking value less than the **sphRegionQuality.qualityInfo**@quality_ranking value of quality ranking sphere region B, quality ranking sphere region A has a higher quality than quality ranking sphere region B. When quality ranking sphere region A partly or entirely overlaps with quality ranking sphere region B, **sphRegionQuality.qualityInfo**@quality_ranking of quality ranking sphere region A shall be equal to **sphRegionQuality.qualityInfo**@quality_ranking of quality ranking sphere region B.
sphRegionQuality.qualityInfo@view_idc	CM	omaf:ViewType	Value 0 indicates that the content is monoscopic, value 1 indicates that the quality ranking sphere region is on the left view of stereoscopic content, value 2 indicates that the quality ranking sphere region is on the right view of stereoscopic content, 3 indicates that the quality ranking sphere region is on both the left and right views. **sphRegionQuality.qualityInfo**@view_idc shall be present when **sphRegionQuality**@view_idc_presence_flag is equal to 1. **sphRegionQuality.qualityInfo**@view_idc shall be absent when **sphRegionQuality**@view_idc_presence_flag is equal to 0.
sphRegionQuality.qualityInfo@orig_width	CM	xs:unsignedShort	Indicates the width of such a monoscopic projected picture for which horRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking sphere region, is equal to 1. Shall not be present when **sphRegionQuality**@quality_type is not equal to 1. Shall be present when **sphRegionQuality**@quality_type is equal to 1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Elements and attributes for SRQR descriptor	Use	Data type	Description
sphRegionQuality.qualityInfo @orig_height	CM	xs:unsignedShort	Indicates the height of such a monoscopic projected picture for which verRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking sphere region, is equal to 1. Shall not be present when **sphRegionQuality**@quality_type is not equal to 1. Shall be present when **sphRegionQuality**@quality_type is equal to 1.
sphRegionQuality.qualityInfo @centre_azimuth	CM	omaf:Range1	Specifies the azimuth of the centre point of the quality ranking sphere region, in units of 2^{-16} degrees, relative to the global coordinate axes. **sphRegionQuality.qualityInfo**@centre_azimuth shall be present when **sphRegionQuality**@remaining_area_flag is equal to 0. **sphRegionQuality.qualityInfo**@centre_azimuth shall be absent in only one **sphRegionQuality.qualityInfo** element and shall be present in all the other **sphRegionQuality.qualityInfo** elements when **sphRegionQuality**@remaining_area_flag is equal to 1.
sphRegionQuality.qualityInfo @centre_elevation	CM	omaf:Range2	Specifies the pitch of the centre point of the quality ranking sphere region, in units of 2^{-16} degrees, relative to the global coordinate axes. s**phRegionQuality.qualityInfo**@centre_elevation shall be present when **sphRegionQuality**@remaining_area_flag is equal to 0. **sphRegionQuality.qualityInfo**@centre_elevation shall be absent in only one **sphRegionQuality.qualityInfo** element and shall be present in all the other **sphRegionQuality.qualityInfo** elements when **sphRegionQuality**@remaining_area_flag is equal to 1.
sphRegionQuality.qualityInfo @centre_tilt	CM	omaf:Range1	Specifies the tilt angle for the quality ranking sphere region in units of 2^{-16} degrees. **sphRegionQuality.qualityInfo**@centre_tilt shall be present when **sphRegionQuality**@remaining_area_flag is equal to 0. **sphRegionQuality.qualityInfo**@centre_tilt shall be absent one **sphRegionQuality.qualityInfo** element and shall be present in all the other **sphRegionQuality.qualityInfo** elements when **sphRegionQuality**@remaining_area_flag is equal to 1.
sphRegionQuality.qualityInfo @azimuth_range	CM	omaf:HRange	Specifies the azimuth range of the quality ranking sphere region through its centre point in units of 2^{-16} degrees. **sphRegionQuality.qualityInfo**@azimuth_range shall be present when **sphRegionQuality**@remaining_area_flag is equal to 0. **sphRegionQuality.qualityInfo**@azimuth_range shall be absent in only one **sphRegionQuality.qualityInfo** element and shall be present in all the other **sphRegionQuality.qualityInfo** elements when **sphRegionQuality**@remaining_area_flag is equal to 1.
sphRegionQuality.qualityInfo @elevation_range	CM	omaf:VRange	Specifies the elevation range of the quality raking sphere region through its centre point in units of 2^{-16} degrees. s**phRegionQuality.qualityInfo**@elevation_range shall be present when **sphRegionQuality**@remaining_area_flag is equal to 0. **sphRegionQuality.qualityInfo**@elevation_range shall be absent in only one **sphRegionQuality.qualityInfo** element and shall be present in all the other **sphRegionQuality.qualityInfo** elements when **sphRegionQuality**@remaining_area_flag is equal to 1.

NOTE 4 A player is recommended to parse spherical region-wise quality ranking (SRQR) descriptors and select the Adaptation Sets and Representations that matches the user's viewing orientation in a manner that:

- The quality ranking value on the region covering the viewport is greater than 0 and less than that for other regions.
- The resolution of the region covering the viewport is suitable for the display. If **sphRegionQuality**@quality_type is equal to 1, **sphRegionQuality.qualityInfo**@orig_width and **sphRegionQuality.qualityInfo**@orig_height represent the width and height of the monoscopic projected picture from which the packed region covering the viewport has been extracted. Otherwise, width and height of VisualSampleEntry could be used to conclude the resolution on the viewport.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

The data types for various elements and attributes shall be as defined in the XML schema. An XML schema for SRQR is defined as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:omaf:2017` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">

    <xs:element name="sphRegionQuality" type="omaf:SphRegionQualityType"/>
    <xs:complexType name="SphRegionQualityType">
        <xs:sequence>
            <xs:element name="qualityInfo" type="omaf:QualityInfoType"
minOccurs="1" maxOccurs="255"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="shape_type" type="xs:unsignedByte" use="optional"
default="0"/>
        <xs:attribute name="remaining_area_flag" type="xs:boolean" use="optional"
default="0"/>
        <xs:attribute name="view_idc_presence_flag" type="xs:boolean"
use="optional" default="0"/>
       <xs:attribute name="quality_ranking_local_flag" type="xs:boolean"
      use="optional" default="0"/>
        <xs:attribute name="quality_type" type="omaf:QualityType" use="required"/>
        <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="QualityInfoType">
                <xs:attribute name="quality_ranking" type="xs:unsignedByte"
use="required"/>
                <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
                <xs:attribute name="orig_width" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="orig_height" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="centre_azimuth" type="omaf:Range1"
use="optional"/>
                <xs:attribute name="centre_elevation" type="omaf:Range2"
use="optional"/>
                <xs:attribute name="centre_tilt" type="omaf:Range1" use="optional"/>
                <xs:attribute name="azimuth_range" type="omaf:HRange"
use="optional"/>
                <xs:attribute name="elevation_range" type="omaf:VRange"
use="optional"/>
                <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:simpleType name="Range1">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="-11796480"/>
            <xs:maxInclusive value="11796479"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Range2">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="-5898240"/>
            <xs:maxInclusive value="5898240"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="HRange">
        <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="23592960"/>
        </xs:restriction>
    </xs:simpleType>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
        <xs:simpleType name="VRange">
            <xs:restriction base="xs:unsignedInt">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="11796480"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="QualityType">
            <xs:restriction base="xs:unsignedByte">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="15"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="ViewType">
            <xs:restriction base="xs:unsignedByte">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="3"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:schema>
```

8.3.6 Signalling of 2D region-wise quality ranking

A **SupplementalProperty** element with a @schemeIdUri attribute equal to "urn:mpeg:mpegI:omaf:2017:2dqr" is referred to as a 2D region-wise quality ranking (2DQR) descriptor.

At most one 2DQR descriptor may be present at Adaptation Set level. At most one 2DQR descriptor may be present at Representation level. A 2DQR descriptor shall not be present at MPD level.

The 2DQR descriptor indicates a quality ranking value of a quality ranking 2D region relative to

- other quality ranking 2D regions in the same Adaptation Set, and
- when **twoDRegionQuality**@quality_ranking_local_flag is equal to 0, 2DQR descriptors with **twoDRegionQuality**@quality_ranking_local_flag equal to 0 and @qualityRanking values in all Adaptation Sets that have the same @value in the DASH **Viewpoint** element as the Adaptation Set containing this 2DQR descriptor or containing the Representation that contains this 2DQR descriptor.

NOTE 1 As specified in ISO/IEC 23009-1, the handling of the **Viewpoint** element is recommended to be applied equally for recognized and unrecognized @schemeIdUri values. It is suggested to use the same @schemeIdUri value for all **Viewpoint** elements in all Adaptation Sets of the same omnidirectional audio-visual content within the same Period.

When the quality ranking value **twoDRegionQuality.twoDqualityInfo**@quality_ranking is non-zero, the picture quality within the entire indicated quality ranking 2D region is approximately constant.

When the 2DQR descriptor that applies to a Representation is present and a 2DRegionQualityRankingBox is present in the track corresponding to the Representation, the 2DQR descriptor shall carry equivalent information as the 2DRegionQualityRankingBox.

The @value attribute of the 2DQR descriptor shall not be present. The 2DQR descriptor shall include a **twoDRegionQuality** element with its sub-elements and attributes as specified in Table 16.

Table 16 – Semantics of elements and attributes of 2DQR descriptor

Elements and attributes for 2DQR descriptor	Use	Data type	Description
twoDRegionQuality	1	omaf:twoDRegionQualityType	Container element which includes one or more 2D region quality information elements (**twoDRegionQuality.twoDqualityInfo**) and common set of attributes (**twoDRegionQuality**@remaining_area_flag, **twoDRegionQuality**@view_idc_presence_flag, **twoDRegionQuality**@quality_ranking_local_flag, **twoDRegionQuality**@quality_type, **twoDRegionQuality**@default_view_idc) that apply to all those quality information elements.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

twoDRegionQuality@remaining_area_flag	O	xs:boolean	Value 0 specifies that all the quality ranking 2D regions are specified by the signalled **twoDRegionQuality.twoDqualityInfo** elements. Value 1 specifies that all except the last quality ranking 2D regions are specified by the signalled **twoDRegionQuality.twoDqualityInfo** elements, and the last remaining quality ranking 2D region is the 2D region within the content coverage, not covered by the union of the quality ranking 2D regions specified by the signalled **twoDRegionQuality.twoDqualityInfo** elements. When not present **twoDRegionQuality**@remaining_area_flag is inferred to be equal to 0. The last remaining quality ranking 2D region may be on both the left and right views. NOTE 2 When **twoDRegionQuality**@remaining_area_flag is equal to 1, the **twoDqualityInfo** element is present for the last quality ranking 2D region but excludes @left_offset, @top_offset, @region_width, and @region_height attributes.
twoDRegionQuality@view_idc_presence_flag	O	xs:boolean	Value 0 specifies that **twoDRegionQuality.twoDqualityInfo**@view_idc is not signalled. Value 1 specifies that **twoDRegionQuality.twoDqualityInfo**@view_idc is signalled and indicates the association of quality ranking 2D regions with particular (left or right or both) views or monoscopic content. When not present **twoDRegionQuality**@view_idc_presence_flag is inferred to be equal to 0.
twoDRegionQuality@quality_ranking_local_flag	O	xs:boolean	Value 0 specifies that the quality ranking information provided in this instance of this descriptor is relative to the quality ranking information provided in all instances of this descriptor with **twoDRegionQuality**@quality_ranking_local_flag equal to 0 and in all instances of @qualityRanking in all Adaptation Sets that have the same @value in the DASH **Viewpoint** element as this Adaptation Set. Value 1 specifies that the quality ranking information provided in this instance of this descriptor is relative to the quality ranking information provided in all instances of this descriptor in this Adaptation Set only. When not present, the value of **twoDRegionQuality**@quality_ranking_local_flag is inferred to be equal to 0. NOTE 3 **twoDRegionQuality**@quality_ranking_local_flag equal to 1 might be used in the Main Adaptation Set of a Preselection and **twoDRegionQuality**@quality_ranking_local_flag equal to 0 could be used in Adaptation Sets that are not the main Adaptation Sets of a Preselection.
twoDRegionQuality@quality_type	M	omaf:QualityType	Indicates which factor causes the differences in the quality of packed regions on the picture. Value 0 specifies that all packed regions correspond to the same projected picture resolution. Value 1 specifies that at least one horRatio value, as derived in subclause 5.4.2, may differ from other horRatio values among all pairs of packed and projected regions of the picture or at least one verRatio value, as derived in subclause 5.4.2, may differ from other verRatio values among all pairs of packed and projected regions of the picture. Values greater 1 are reserved.
twoDRegionQuality@default_view_idc	CM	omaf:ViewType	Value 0 indicates that all the quality ranking 2D regions are monoscopic. Value 1 indicates that all the quality ranking 2D regions are on the left view of stereoscopic content. Value 2 indicates that all the quality ranking 2D regions are on the right view of stereoscopic content. Value 3 indicates that all the quality ranking 2D regions are on both the left and right views. **twoDRegionQuality**@default_view_idc shall be present when **twoDRegionQuality**@view_idc_presence_flag is equal to 0. **twoDRegionQuality**@default_view_idc shall be absent when **twoDRegionQuality**@view_idc_presence_flag is equal to 1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

twoDRegionQuality.twoDqualityInfo	1..255	omaf:twoDQualityInfoType	Element whose attribute **twoDRegionQuality.twoDqualityInfo**@quality_ranking provides quality ranking for one quality ranking 2D region described by its attributes **twoDRegionQuality.twoDqualityInfo**@view_idc, **twoDRegionQuality.twoDqualityInfo**@left_offset, **twoDRegionQuality.twoDqualityInfo**@top_offset, **twoDRegionQuality.twoDqualityInfo**@region_width, **twoDRegionQuality.twoDqualityInfo**@region_height.
twoDRegionQuality.twoDqualityInfo@quality_ranking	M	xs:unsignedByte	Specifies a quality ranking value of the quality ranking 2D region. **twoDRegionQuality.twoDqualityInfo**@quality_ranking equal to 0 indicates that the quality ranking is not defined. When quality ranking 2D region A has a non-zero **twoDRegionQuality.twoDqualityInfo**@quality_ranking value less than the **twoDRegionQuality.twoDqualityInfo**@quality_ranking value of quality ranking 2D region B, quality ranking 2D region A has a higher quality than quality ranking 2D region B. When quality ranking 2D region A partly or entirely overlaps with quality ranking 2D region B, **twoDRegionQuality.twoDqualityInfo**@quality_ranking of quality ranking 2D region A shall be equal to **twoDRegionQuality.twoDqualityInfo**@quality_ranking of quality ranking 2D region B.
twoDRegionQuality.twoDqualityInfo@view_idc	CM	omaf:ViewType	Value 0 indicates that the content is monoscopic, value 1 indicates that the quality ranking 2D region is on the left view of stereoscopic content, value 2 indicates that the quality ranking 2D region is on the right view of stereoscopic content, 3 indicates that the quality ranking 2D region is on both the left and right views. **twoDRegionQuality.twoDqualityInfo**@view_idc shall be present when **twoDRegionQuality**@view_idc_presence_flag is equal to 1. **twoDRegionQuality.twoDqualityInfo**@view_idc shall be absent when **twoDRegionQuality**@view_idc_presence_flag is equal to 0.
twoDRegionQuality.twoDqualityInfo@orig_width	CM	xs:unsignedShort	Indicates the width of such a monoscopic projected picture for which horRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking sphere region, is equal to 1. Shall not be present when **twoDRegionQuality**@quality_type is not equal to 1. Shall be present when **twoDRegionQuality**@quality_type is equal to 1.
twoDRegionQuality.twoDqualityInfo@orig_height	CM	xs:unsignedShort	Indicates the height of such a monoscopic projected picture for which verRatio, as derived in subclause 5.4.2 for each of the packed regions that cover the quality ranking sphere region, is equal to 1. Shall not be present when **twoDRegionQuality**@quality_type is not equal to 1. Shall be present when **twoDRegionQuality**@quality_type is equal to 1.
twoDRegionQuality.twoDqualityInfo@left_offset	CM	xs:unsignedShort	Specifies the horizontal coordinate of the upper left corner of the quality ranking 2D region within the picture in units of luma samples. **twoDRegionQuality.twoDqualityInfo**@left_offset shall be present when **twoDRegionQuality**@remaining_area_flag is equal to 0. **twoDRegionQuality.twoDqualityInfo**@left_offset shall be absent in only one **twoDRegionQuality.twoDqualityInfo** element and shall be present in all the other **twoDRegionQuality.twoDqualityInfo** elements when **twoDRegionQuality**@remaining_area_flag is equal to 1.
twoDRegionQuality.twoDqualityInfo@top_offset	CM	xs:unsignedShort	Specifies the vertical coordinate of the upper left corner of the quality ranking 2D region within the picture in units of luma samples. **twoDRegionQuality.twoDqualityInfo**@top_offset shall be present when **twoDRegionQuality**@remaining_area_flag is equal to 0. **twoDRegionQuality.twoDqualityInfo**@top_offset shall be absent in only one **twoDRegionQuality.twoDqualityInfo** element and shall be present in all the other **twoDRegionQuality.twoDqualityInfo** elements when **twoDRegionQuality**@remaining_area_flag is equal to 1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

twoDRegionQuality.twoDqualityInfo@region_width	CM	xs:unsignedShort	Specifies the width of the quality ranking 2D region within the picture in units of luma samples. **twoDRegionQuality.twoDqualityInfo**@region_width shall be present when **twoDRegionQuality**@remaining_area_flag is equal to 0. **twoDRegionQuality.twoDqualityInfo**@region_width shall be absent in only one **twoDRegionQuality.twoDqualityInfo** element and shall be present in all the other **twoDRegionQuality.twoDqualityInfo** elements when **twoDRegionQuality**@remaining_area_flag is equal to 1.
twoDRegionQuality.twoDqualityInfo@region_height	CM	xs:unsignedShort	Specifies the height of the quality ranking 2D region within the picture in units of luma samples. **twoDRegionQuality.twoDqualityInfo**@region_height shall be present when **twoDRegionQuality**@remaining_area_flag is equal to 0. **twoDRegionQuality.twoDqualityInfo**@region_height shall be absent in only one **twoDRegionQuality.twoDqualityInfo** element and shall be present in all the other **twoDRegionQuality.twoDqualityInfo** elements when **twoDRegionQuality**@remaining_area_flag is equal to 1.

NOTE 4 A player is recommended to parse 2D region-wise quality ranking (2DQR) descriptors and select the Adaptation Sets and Representations that matches the user's viewing orientation in a manner that:

- The quality ranking value on the region covering the viewport is greater than 0 and less than that for other regions.
- The resolution of the region covering the viewport is suitable for the display. If **twoDRegionQuality**@quality_type is equal to 1, **twoDRegionQuality.twoDqualityInfo**@orig_width and **twoDRegionQuality.twoDqualityInfo**@orig_height represent the width and height of the monoscopic projected picture from which the packed region covering the viewport has been extracted. Otherwise, width and height of VisualSampleEntry could be used to conclude the resolution on the viewport.

The data types for various elements and attributes shall be as defined in the XML schema. An XML schema for 2DQR is defined as shown below. The schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:omaf:2017 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">

    <xs:element name="twoDRegionQuality" type="omaf:twoDRegionQualityType"/>
    <xs:complexType name="twoDRegionQualityType">
        <xs:sequence>
            <xs:element name="twoDqualityInfo" type="omaf:twoDQualityInfoType"
minOccurs="1" maxOccurs="255"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="remaining_area_flag" type="xs:boolean" use="optional"
default="0"/>
        <xs:attribute name="view_idc_presence_flag" type="xs:boolean"
use="optional" default="0"/>
       <xs:attribute name="quality_ranking_local_flag" type="xs:boolean"
      use="optional"   default="0"/>
        <xs:attribute name="quality_type" type="omaf:QualityType" use="required"/>
        <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="twoDQualityInfoType">
                <xs:attribute name="quality_ranking" type="xs:unsignedByte"
use="required"/>
                <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
                <xs:attribute name="orig_width" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="orig_height" type="xs:unsignedShort"
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
use="optional"/>
                <xs:attribute name="left_offset" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="top_offset" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="region_width" type="xs:unsignedShort"
use="optional"/>
                <xs:attribute name="region_height" type="xs:unsignedShort"
use="optional"/>
                <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
   <xs:simpleType name="QualityType">
        <xs:restriction base="xs:unsignedByte">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="15"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ViewType">
        <xs:restriction base="xs:unsignedByte">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="3"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

8.3.7 Signalling of fisheye omnidirectional video

A **`SupplementalProperty`** element with a `@schemeIdUri` attribute equal to `"urn:mpeg:mpegI:omaf:2017:fomv"` is referred to as a fisheye omnidirectional video (FOMV) descriptor.

At most one FOMV descriptor may be present at Adaptation Set level. An FOMV descriptor shall not be present at MPD or Representation level.

The FOMV descriptor indicates that each Representation carries a fisheye omnidirectional video track containing a `FisheyeOmniVideoBox`. The `@value` attribute of the FOMV descriptor shall not be present. The FOMV descriptor shall include an `omaf:@view_dimension_idc` attribute whose value shall be as specified in Table 17.

Table 17 – Semantics of `omaf:@view_dimension_idc` attribute

Attribute for FOMV descriptor	Use	Data type	Description
`omaf:@view_dimension_idc`	M	omaf:viewDIdcType	Has the same semantics as the `view_dimension_idc` syntax element (as specified in subclause 6.2.2) of the `FisheyeVideoEssentialInfoStruct` syntax structure in the `FisheyeOmniVideoBox` in the tracks carried in the Representations of this Adaptation Set. For ISO base media file format Segments, `view_dimension_idc` shall be equal to `view_dimension_idc` in `FisheyeVideoEssentialInfoBox` in sample entries of the Initialization Segment.

The data type for the attribute shall be as defined in the XML schema. An XML schema for fisheye omnidirectional video signalling shall be as shown below. The schema shall be represented in an XML schema that has namespace `urn:mpeg:mpegI:omaf:2017` and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    elementFormDefault="qualified">
    <xs:attribute name="view_dimension_idc" type="omaf:viewDIdcType"/>
    <xs:simpleType name="viewDIdcType">
        <xs:restriction base="xs:unsignedByte">
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="7"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

9 Omnidirectional media encapsulation and signalling in MMT

9.1 Architecture of MMT delivery in OMAF

Figure 23 depicts the reference architecture for OMAF content delivery over MMT:

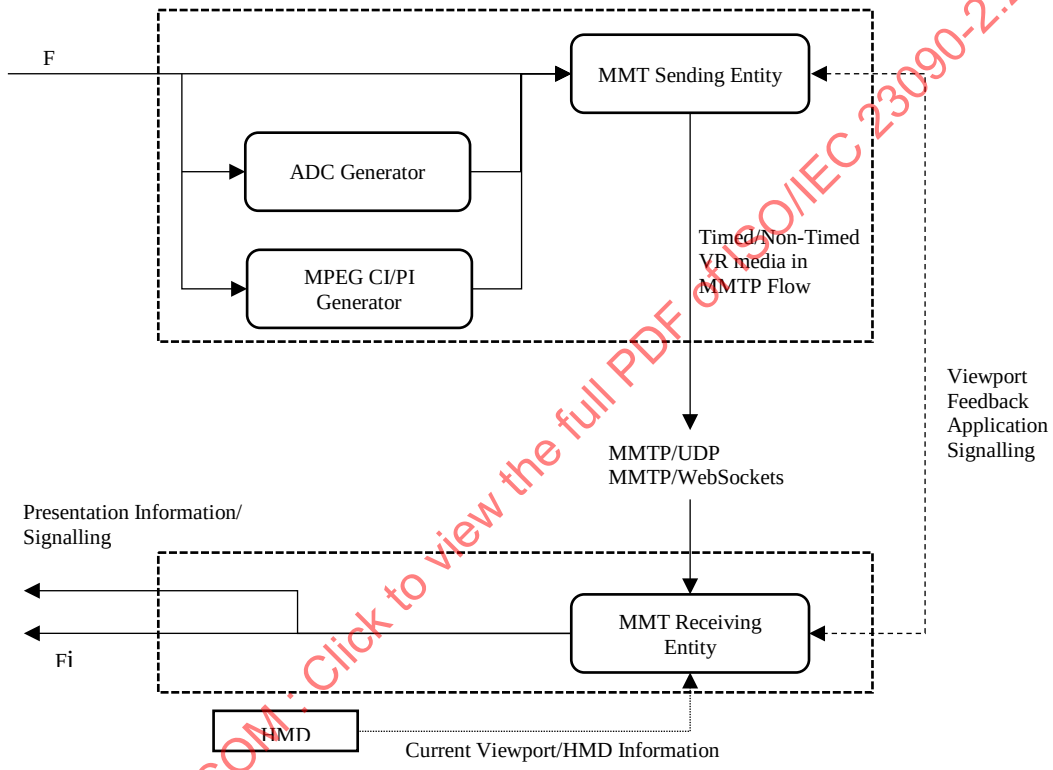


Figure 23 – Reference Architecture for OMAF over MMT

The OMAF content may be described in ADC (as defined in ISO/IEC 23008-1:2017, subclause 6.5) to assist the MMT sending entity during the streaming process. The Presentation Information should contain information to describe MPUs that are conformant to OMAF to enable appropriate processing by the application.

MMT delivery may use the MMTP protocol over UDP, MMTP/WebSockets/TCP, or another alternative.

The player receives information about the current viewing direction and viewport and the HMD characteristics. Only a portion of the 360 video is needed at a time depending on the current viewing direction. Field-of-view streaming (FoV) or alternatively called as view-dependent streaming is used to reduce the bandwidth needs during streaming. When MMT is used, view-dependent streaming may be achieved using one of the following two approaches:

– Client-based approach: the MMT receiving entity may be instructed by the player to select a subset of the Assets that carry video data covering the current viewport. MMT session control procedures as defined in ISO/IEC 23008-1 shall be used to request the selected set of Assets from the MMT sending entity. The player uses the VR application specific signalling message to convey the content coverage and the region-wise packing information in accordance with the definitions in subclauses 7.5.3, 7.6 and 9.3, to select appropriate asset to switch to for view-dependent streaming.

– Server-based approach: the MMT receiving entity relies on the MMT sending entity to select the correct subset of assets that provide video content to cover the current viewport. The receiving entity uses the VR application-specific signalling as described in subclause 9.3 to send information about the current viewport to the sending entity.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

9.2 OMAF signalling in MPEG composition information

MMT supports MPEG CI as the presentation information that describes the media presentation. In MPEG CI, OMAF assets shall be mapped to a to an HTML5 canvas element instead of an HTML5 video element as shown in the following example:

```
<MediaSync begin="T10M5S" dur="T32M" refId="canvas1">
      <sourceList componentId="VRvideo">
            <mediaSrc
mimeType='video/mp4;codecs="encv.cbcs.resv.podv+erpv.hvc1"'>
media1.mp4</mediaSrc>
      </sourceList>
</MediaSync>
```

The `codecs` MIME type parameter shall be set according to subclause 7.1.8.

The user agent should decode the OMAF asset in the background, e.g. using a hidden HTML5 video element and render the omnidirectional media to the referenced HTML5 canvas element.

For view-dependent media streaming and playback, the media data should be retrieved using a protocol that supports view-dependent streaming of VR content, such as using MMT with VR specific signalling (see subclause 9.3). In such case, an MMT session entry point should be used as the mediaSrc element.

9.3 VR application-specific MMT signalling

9.3.1 General

MMT, specified in ISO/IEC 23008-1, defines an application-specific signalling message that allows for the delivery of application-specific information. For the purpose of streaming VR content that is formatted according to the OMAF specification, a VR specific Asset descriptor, a stereo video Asset descriptor, and a VR application-specific signalling message are defined. The VR application-specific signalling message shall have an application identifier with a URN of value of "`urn:mpeg:mmt:app:vr:2017`" that is specified in ISO/IEC 23008-1.

A new asset descriptor for OMAF VR formatted content is defined under the name VR information asset descriptor. The VR information descriptor shall be present in all assets that carry OMAF formatted content. A stereo video asset descriptor is also defined, and shall be present in all assets that carry OMAF VR formatted stereoscopic content.

In the specified VR signalling message, the following set of application message types are defined:

- VRViewDependentSupportQuery: the client uses this command to discover if the server supports view-dependent streaming.
- VRViewDependentSupportResponse: the server replies with an indication of its support capability for view-dependent streaming.
- VRViewportChangeFeedback: the receiving entity sends an indication of the current viewport to the sending entity.
- VRViewDependentAssetsInformation: upon determining the set of OMAF assets that match the requested viewport, the sending entity sends this message to inform the client about the new OMAF assets that will be streamed to the receiving entity.

The VRViewportChangeFeedback and VRViewDependentAssetsInformation message are used together to support server-driven view-dependent streaming of OMAF assets. The content selection message is used to support the client-driven view-dependent streaming scenario, respectively.

To support guided rendering, where the renderer follows an indicated Region of Interest, or to follow the recommended viewport timed metadata track of OMAF, the VRROIGuide application message type is defined.

The list of defined application message types is provided in Table 18.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Table 18 – VR application message types

Application message type	Application message name
0x01	VRViewDependentSupportQuery
0x02	VRViewDependentSupportResponse
0x03	VRViewportChangeFeedback
0x04	VRViewDependentAssetInformation
0x05	VRROIGuide
0x06	VR3DAudioAssetInformation
0x07-0xFF	Reserved for future use

9.3.2 MMT signalling

9.3.2.1 VR information asset descriptor

9.3.2.1.1 General

This asset descriptor is used to inform the receiving entity and the VR application about the content of the current asset that carries VR content. The information describes the projection type that is used, how the VR content is region-wise frame packed, and what areas on the sphere it covers. The indication if content is stereoscopic with frame packing is provided through a separate asset descriptor.

9.3.2.1.2 Syntax

The syntax of the VR information asset descriptor is shown in Table 19.

Table 19 – VR information asset descriptor syntax

Syntax	Value	No. of bits	Mnemonic
VR_information_descriptor() {			
descriptor_tag		16	uimsbf
descriptor_length		16	uimsbf
rwfp_flag		1	bslbf
srqr_flag		1	bslbf
2dqr_flag		1	bslbf
reserved	'1 1111'	5	bslbf
ProjectionFormatStruct()			
InitialViewingOrientationSample()			
ContentCoverageStruct()			
if(rwfp_flag == 1) {			
RegionWisePackingStruct()			
}			
if(srqr_flag == 1) {			
SphereRegionQualityRankingStruct()			
}			
if(2dqr_flag == 1) {			
2DRegionQualityRankingStruct()			
}			
}			

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

9.3.2.1.3 Semantics

`descriptor_tag` indicates the type of a descriptor.

`descriptor_length` specifies the length in bytes counting from the next byte after this field to the last byte of the descriptor.

`rwfp_flag` equal to 1 indicates that region-wise frame packing has been applied to the content of this asset and that the RegionWisePackingStruct that describes it is present.

`srqr_flag` equal to 1 indicates that sphere region quality information is present.

`2dqr_flag` equal to 1 indicates that 2D region quality information is present.

`ProjectionFormatStruct`() provides information on the projection format that is used. `ProjectionFormatStruct`() is identical to the definition in subclause 7.5.2.

`InitialViewingOrientationSample`() provides information about the current initial viewing orientation. `InitialViewingOrientationSample`() is identical to the definition in subclause 7.7.4.

`ContentCoverageStruct`() indicates the sphere region(s) covered by the track. `ContentCoverageStruct`() is identical to the definition in subclause 7.5.5.

`RegionWisePackingStruct`() indicates that the projected pictures are packed region-wise and require unpacking prior to rendering, according to the region-wise packing process information as indicated. `RegionWisePackingStruct()` is identical to the definition in subclause 7.5.3.

`SphereRegionQualityRankingStruct`() indicates a relative quality order of quality ranking sphere regions. `SphereRegionQualityRankingStruct()` is identical to the definition in subclause 7.8.2.

`2DRegionQualityRankingStruct`() indicates a relative quality order of quality ranking 2D regions. `2DRegionQualityRankingStruct()` is identical to the definition in subclause 7.8.3.

9.3.2.2 VRViewDependentSupportQuery

9.3.2.2.1 Syntax

The syntax of the VRViewDependentSupportQuery is shown in Table 20.

Table 20 – VRViewDependentSupportQuery

Syntax	Value	No. of bits	Mnemonic
`Application() {`			
`  message_id`		16	uimsbf
`  version`		8	uimsbf
`  length`		16	uimsbf
`  message_payload{`			
`  application_identifier()`			
`  if(application_identifier == "urn:mpeg:mmt:app:vr:2017")`			
`  {`			
`  app_message_type`		8	uimsbf
`  if(app_message_type == 0x01) {`			
`  hmd_hor_resolution`		16	uimsbf
`  hmd_ver_resolution`		16	uimsbf
`  hmd_hor_fov`		32	uimsbf
`  hmd_ver_fov`		32	uimsbf
`  }`			
`  }`			
`  }`			
`}`			

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

9.3.2.2.2 Semantics

`message_id` indicates the identifier of the VRViewDependentSupportQuery message.

`version` indicates the version of VRViewDependentSupportQuery message.

`length` indicates the length of VRViewDependentSupportQuery message in bytes, counting from the beginning of the next field to the last byte of the VRViewDependentSupportQuery message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

`hmd_hor_resolution` and `hmd_ver_resolution` provides the horizontal and vertical resolution of the display of the HMD in units of square pixels.

`hmd_hor_fov` and `hmd_ver_fov` provide the horizontal and vertical field of view of the HMD, in units of 2^{-16} degrees. `hmd_hor_fov` shall be in the range of 0 to 360 * 2^{16}, inclusive. `hmd_ver_fov` shall be in the range of 0 to 180 * 2^{16}, inclusive.

9.3.2.3 VRViewDependentSupportResponse

9.3.2.3.1 Syntax

The syntax of the ViewDependentSupportResponse is shown in Table 21.

Table 21 – VRViewDependentSupportResponse syntax

Syntax	Value	No. of bits	Mnemonic
`Application() {`			
`  message_id`		16	uimsbf
`  version`		8	uimsbf
`  length`		16	uimsbf
`  message_payload{`			
`  application_identifier()`			
`  if(application_identifier == "urn:mpeg:mmt:app:vr:2017")`			
`  {`			
`  app_message_type`		8	uimsbf
`  if(app_message_type == 0x02) {`			
`  view_dependent_support`		1	bslbf
`  reserved`	'1111111'	7	uimsbf
`  }`			
`  }`			
`  }`			
`}`			

9.3.2.3.2 Semantics

`message_id` indicates the identifier of the VRViewDependentSupportResponse message.

`version` indicates the version of VRViewDependentSupportResponse message.

`length` indicates the length of VRViewDependentSupportResponse message in bytes, counting from the beginning of the next field to the last byte of the VRViewDependentSupportResponse message. The value of this field shall not be equal to 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`application_identifier` indicates the application identifier as a urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

`view_dependent_support` equal to 1 indicates that view-dependent streaming is supported by the server. `view_dependent_support` equal to 0 indicates that view dependent streaming is not supported by the server.

9.3.2.4 VRViewportChangeFeedback

9.3.2.4.1 General

The MMT VR receiving entity feedbacks the virtual camera direction information periodically or in case of FOV changing event to the MMT sending entity to inform about the current VR virtual camera direction.

9.3.2.4.2 Syntax

The syntax of the VRViewportChangeFeedback is shown in Table 22.

Table 22 – VRViewportChangeFeedback

Syntax	Value	No. of bits	Mnemonic
`Application() {`			
`  message_id`		16	uimsbf
`  version`		8	uimsbf
`  length`		16	uimsbf
`  message_payload{`			
`  application_identifier()`			
`  if(application_identifier == "urn:mpeg:mmt:app:vr:2017")`			
`  {`			
`  app_message_type`		8	uimsbf
`  if(app_message_type == 0x03) {`			
`  dirx`		16	uimsbf
`  diry`		16	uimsbf
`  dirz`		16	uimsbf
`  last_processed_media_timestamp`		64	uimsbf
`  }`			
`  }`			
`  }`			
`}`			

9.3.2.4.3 Semantics

`message_id` indicates the identifier of the VRViewportChangeFeedback message.

`version` indicates the version of VRViewportChangeFeedback message.

`length` indicates the length of VRViewportChangeFeedback message in bytes, counting from the beginning of the next field to the last byte of the VRViewportChangeFeedback message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`dirx`, `diry`, and `dirz` define the x, y, and z component, respectively, of the three-dimensional viewing direction unit vector in a Cartesian coordinate system with (x, y, z) equal to (1, 0, 0) corresponding to the sphere location with (ϕ, θ) equal to (0, 0). The value of `dirx`, `diry`, or `dirz` shall be in the range of 1 to 65535, inclusive, where 1 corresponds to −1, 32768 corresponds to 0, and 65535 corresponds to +1.

The fields `dirx`, `diry`, and `dirz` may be calculated according to the azimuth ϕ and elevation θ, which may be obtained from HMD sensor. Herein, the viewing direction denotes a three-dimensional vector from the centre of the sphere pointing to a location on the surface of the sphere. An example of specifying `dirx`, `diry`, and `dirz` is given in clause F.2.

`last_processed_media_timestamp` indicates the presentation timestamp of the last media unit that has been appended to the decoder buffer. This field is used by the MMT sending entity to determine the next media unit from the new asset that is sent to the OMAF player. The next media unit is the one with a timestamp or sequence number immedialy following the indicated timestamp. The MMT sending entity switches from transmitting the old asset (representing the old viewport) to transmitting the new asset (representing the new viewport) starting from the following media timestamp, in order to reduce the delay of receiving the new viewport. This means that the transmitted media data from the new asset may overlap in media time with already transmitted media data from the old asset. If more than one OMAF asset is being delivered, the minimum value shall be used.

9.3.2.5 VRViewDependentAssetInformation

9.3.2.5.1 General

This signalling is used to indicate to the receiving entity that MMT packets belonging to the new asset as specified by the packet_id will be sent to the receiving entity from the sending entity. Upon receiving this signal, the next fragment received by the client belongs to the new Asset as indicated by the packet_id. Based on the asset ID, the renderer is able to correctly associate the received media with the coverage information and to correctly render the omnidirectional video using the video texture from the new asset.

9.3.2.5.2 Syntax

The syntax of the VRViewDependentAssetInformation is shown in Table 23.

Table 23 – VRViewDependentAssetInformation syntax

Syntax	Value	No. of bits	Mnemonic
Application() {			
message_id		**16**	**uimsbf**
version		**8**	**uimsbf**
length		**16**	**uimsbf**
message_payload{			
application_identifier()			
if(application_identifier == "urn:mpeg:mmt:app:vr:2017")			
{			
app_message_type		**8**	**uimsbf**
if(app_message_type == 0x04) {			
reserverd	**'1111 1111'**	**8**	**bslbf**
packet_id		**16**	**uimsbf**
ContentCoverageStruct()			
}			
}			
}			
}			

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

9.3.2.5.3 Semantics

`message_id` indicates the identifier of the VRViewDependentAssetInformation message.

`version` indicates the version of VRViewDependentAssetInformation message.

`length` indicates the length of VRViewDependentAssetInformation message in bytes, counting from the beginning of the next field to the last byte of the VRViewDependentAssetInformation message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

`packet_id` indicates the packet_id that is associated with the OMAF asset. This field is passed to the renderer to indicate the asset ID of the asset that is currently being received and played in high resolution. Based on the asset ID, the renderer is able to determine the coverage information of the video texture that corresponds to the current asset.

`ContentCoverageStruct()` indicates the sphere region(s) covered by the content. `ContentCoverageStruct()` is identical to the definition in subclause 7.5.5.

9.3.2.6 VRROIGuide

9.3.2.6.1 General

MMT sending entity sends VRROIGuide message to guide what to display from the delivered entire video content. If the MMT receiving entity receives the media data of the entire video, only the region specified by the VRROIGuide message shall be presented.

9.3.2.6.2 Syntax

The syntax of VRROIGuide message is defined in Table 24.

Table 24 – VRROIGuide syntax

Syntax	Value	No. of bits	Mnemonic
`Application() {`			
`  message_id`		16	uimsbf
`  version`		8	uimsbf
`  length`		16	uimsbf
`  message_payload{`			
`  application_identifier()`			
`  if(application_identifier == "urn:mpeg:mmt:app:vr:2017")`			
`  {`			
`  app_message_type`		8	uimsbf
`  if(app_message_type == 0x05) {`			
`  guide_type`		1	uimsbf
`  reserved`	'1111 1111'	7	bslbf
`  region_type`		8	uimsbf
`  guide_region(){`			
`  if ((region_type == 0x01) || (region_type == 0x03)) {`			
`  guide_shape_type`		8	uimsbf
`  SphereRegionStruct(1)`			
`  }`			

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

else if ((region_type == 0x02) || (region_type == 0x04)) {			
centre_x		16	uimsbf
centre_y		16	uimsbf
width		16	uimsbf
height		16	uimsbf
}			
}			
guide_start		32	uimsbf
guide_duration		32	uimsbf
}			
}			
}			

9.3.2.6.3 Semantics

`message_id` indicates the identifier of the VRROIGuide message.

`version` indicates the version of VRROIGuide message.

`length` indicates the length of VRROIGuide message in bytes, counting from the beginning of the next field to the last byte of the VRROIGuide message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as a urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

`guide_type` indicates whether presenting only the guide region specified by `guide_region` is mandatory. If this flag is set to 1, only the guide region should be displayed. If this flag is set to 0, both the guide region and the rest of region may be displayed.

`region_type` defines the type of guide region. The values for this field are specified in Table 25.

Table 25 – Value of `region_type`

Type	Description
0x00	reserved
0x01	centre point of a viewport on sphere
0x02	centre point of a viewport on frame
0x03	viewport on sphere
0x04	viewport on frame
0x05~0xFF	reserved

`guide_region` specifies the guide region to be displayed. If the value of `region_type` is 0x01 or 0x03, the guide region is specified as the defintion of `SphereRegionStruct` in subclause 7.5.6.

`guide_shape_type` specifies the shape of the guide region. `guide_shape_type` has the same semantics as `shape_type` specified in subclause 7.7.2.3.

`centre_x` specifies the horizontal centre positon of the guide region in pixels.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

`centre_y` specifies the vertical centre positon of the guide region in pixels.

`width` specifies the width of the guide region in pixels.

`height` specifies the height of the guide region in pixels.

`guide_start` indicates the presentation time of media data where the region guide starts. This field is a UTC time in NTP format and 32 bits long.

`guide_duration` indicates the duration the region guide from the time indicated by `guide_start`. This field is expressed in milliseconds.

9.3.2.7 Stereo video asset descriptor

9.3.2.7.1 Syntax

The syntax of the stereo video asset descriptor is shown in Table 26.

Table 26 – Stereo video asset descriptor

Syntax	Value	No. of bits	Mnemonic
`Stereo_video_descriptor(){`			
descriptor_tag		16	uimsbf
descriptor_length		8	uimsbf
StereoVideoBox()			
`}`			

9.3.2.7.2 Semantics

`descriptor_tag` indicates the type of a descriptor.

`descriptor_length` specifies the length in bytes counting from the next byte after this field to the last byte of the descriptor.

`StereoVideoBox()` provides a copy of the `StereoVideoBox` as defined in ISO/IEC 14496-12.

9.3.2.8 VR3DAudioAssetInformation

9.3.2.8.1 General

The VR3DAudioAssetInformation message shall be conveyed for all Assets that carry OMAF formatted content compliant to the OMAF 3D audio baseline profile.

9.3.2.8.2 Syntax

The syntax of **VR3DAudioAssetInformation** message is defined in Table 27.

Table 27 – VR3DAudioAssetInformation syntax

Syntax	Value	No. of bits	Mnemonic
VR3DAudioAssetInformation() {			
message_id		16	uimsbf
version		8	uimsbf
length		16	uimsbf
`message_payload {`			
application_identifier()			

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Syntax	Value	No. of bits	Mnemonic
if(application_identifier=="urn:mpeg:mmt:app:vr:2017")			
{			
app_message_type		8	uimsbf
if(app_message_type == 0x06) {			
number_of_assets		8	uimsbf
for (i=0; i<number_of_assets; i++) {			
asset_id_length		32	uimsbf
for (j=0; j<asset_id_length; j++) {			
asset_id_byte		8	uimsbf
}			
codec_code		4*8	uimsbf
profile_level_indication		8	uimsbf
num_preselections		8	uimsbf
channel_configuration		6	uimbsf
multi_stream_info_present		1	bslbf
reserved	'1'	1	bslbf
for (j=0; j<num_preselections; j++) {			
preselection_id		8	uimbsf
interactivity_enabled		1	bslbf
language_present		1	bslbf
accessibility_role_present		1	bslbf
label_present		1	bslbf
reserved	'1111'	4	bslbf
if (language_present) {			
num_languages_minus1		8	uimbsf
for(k=0;k<num_languages_minus1+1;k++) {			
language_length		8	uimbsf
for (l=0;l< language_length;l++) {			
language_byte		8	uimbsf
}			
}			
}			
if (accessibility_role_present) {			
for(k=0;k<num_languages_minus1+1;k++) {			
accessibility		8	uimbsf
}			
role		8	uimbsf
}			
if (label_present) {			
label_length		8	uimbsf
for (k=0; k< label_length; k++) {			
label_data_byte		8	uimbsf
}			
}			
if (multi_stream_info_present) {			
preselection_aux_stream_info()			Table 9.12
}			
} /* end of for num_preselections loop */			
if (multi_stream_info_present) {			
multi_stream_info()			Table 9.13

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Syntax	Value	No. of bits	Mnemonic
} } /* end of for number_of_assets loop*/ } /* end of if(app_message_type == 0x06)*/ } } }			

9.3.2.8.3 Semantics

`message_id` indicates the identifier of the VR3DAudioAssetInformation message.

`version` indicates the version of VR3DAudioAssetInformation message.

`length` indicates the length of VR3DAudioAssetInformation message in bytes, counting from the beginning of the next field to the last byte of the VR3DAudioAssetInformation message. The value of this field shall not be equal to 0.

`application_identifier` indicates the application identifier as an urn that uniquely identifies the application to consume the contents of this message.

`app_message_type` defines an application-specific message type provided in Table 18.

`number_of_assets` specifies the number of audio assets described by this descriptor.

`asset_id_length` specifies the length in bytes of the audio asset id.

`asset_id_byte` contains a byte of the audio asset id.

`codec_code` specifies the 4-character code for MPEG-H 3D audio. The value of these four characters shall be one of `'mhm1'` or `'mhm2'` with a semantic meaning for these codes as specified in ISO/IEC 23008-3.

`profile_level_indication` indicates the audio profile and level of the associated Preselection and shall contain an `mpegh3daProfileLevelIndication` field as specified in ISO/IEC 23008-3:2015, subclause 5.3.2.

`num_preselections` indicates the number of Preselections that are available within the main stream and all auxiliary streams. The minimum number of `num_preselections` shall be '1' for the main stream. For auxiliary streams `num_preselections` shall have the value '0' so that for auxiliary streams no Preselection information is present in the descriptor. This field contains a `mae_numGroupPresets` field as specified in ISO/IEC 23008-3:2015, subclause 15.3.

`channel_configuration` specifies the channel configuration and shall have the same value as the `ChannelConfiguration` field specified in ISO/IEC 23091-3. Valid values are 1-7,9-12, 14-17 or 19.

`multi_stream_info_present` equal to 1 indicates that the elements in the `multi_stream_info()` structure are present.

`preselection_id` identifies the ID of this Preselection. The first Preselection in the loop shall have the lowest preselection_id and shall be the default Preselection. This field indicates the `mae_GroupPresetID` field as specified in ISO/IEC 23008-3:2015, subclause 15.3.

`interactivity_enabled` equal to 1 indicates that that the audio Preselection contains elements with associated metadata, which enable user interactivity.

`language_present` equal to 1 indicates that language information for this Preselection is present.

`accessibility_role_present` equal to 1 indicates that accessibility and role information for this Preselection is present.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

label_present equal to 1 indicates that a text label for this Preselection is present.

num_languages_minus1 plus 1 specifies the number of languages that are available within this Preselection. When not present the value of num_languages_minus1 shall be inferred to be equal to 0.

language_length specifies the length in bytes of each language supported in the Preselection. The first language in the loop (k is equal to 0) shall be the primary language for the Preselection. The remaining language(s) in the loop (k is not equal to 0) shall indicate the additional language(s) available in the Preselection.

language_byte contains a UTF-8 character of the k-th language of the Preselection. The language of the Preselection shall be given by a language tag as defined by IETF BCP 47. The language indicated by this field should correspond to the information conveyed in mae_contentLanguage of the default dialog element: the maeGroup which is marked as default in mae_switchGroupDefaultGroupID and is tagged in mae_contentKind as dialog. This information is carried in the AudioSceneInformation() of the MPEG-H audio stream as specified in ISO/IEC 23008-3.

accessibility identifies the accessibility support for each language in this Preselection. Table 28 specifies the bit used to indicate if the Preselection contains support for a particular audio accessibility service. When one bit specified in Table 28 is set to '1' it indicates the Preselection contains the corresponding audio accessibility service.

Table 28 – Accessibility bits

Bit	Audio accessibility service
0 (MSB)	For Visually Impaired (Video description service)
1	Dialog enhancement enabled
2	Emergency information
3-7	Reserved zero bits [a]
[a] Reserved bits shall be set to zero.	

The setting of the bits in the accessibility field should correspond to the mae_groupPresetKind value in the mae_GroupPresetDefinition() structure and the mae_contentKind values in the mae_ContentData() structures in the AudioSceneInformation() of the MPEG-H 3D audio stream as specified in ISO/IEC 23008-3. The mapping from the MPEG-H audio metadata fields should be done as follows:

- Bit 0 should be set to '1', if the mae_contentKind value of at least one audio element is set to '9' ("audio description/visually impaired").
- Bit 1 should be set to '1', if at least the dialog audio elements with a mae_contentKind value of '2' ("dialogue") have mae_allowGainInteractivity set to '1' and mae_interactivityMaxGain set to a non-zero value in the corresponding mae_GroupDefinition() structure.
- Bit 2 should be set to '1', if the mae_contentKind value of at least one audio element is set to '12' ("emergency").

role indicates the role or service type of the Preselection. The role values shall correspond to the role scheme @schemeIdUri equal to "urn:mpeg:dash:role:2011" defined in ISO/IEC 23009-1. For a description of the role values, see ISO/IEC 23009-1:2014, subclause 5.8.4.4.

label_length specifies the length in bytes of this Preselection text label.

label_data_byte contains a UTF-8 character of the Preselection text label.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Table 29 – Syntax for preselection_aux_stream_info()

Syntax	Value	No. of bits	Mnemonic
preselection_aux_stream_info() {			
num_preselection_aux_streams		8	uimbsf
for (m=0; m<num_preselection_aux_streams; m++) {			
aux_stream_id		8	uimbsf
}			
}			

num_preselection_aux_streams indicates the number of auxiliary streams that are required for this specific Preselection.

aux_stream_id ientifies the ID of the auxiliary stream that is required for this specific Preselection.

Table 30 – Syntax for multi-stream_info()

Syntax	Value	No. of bits	Mnemonic
multi_stream_info() {			
this_is_main_stream		1	bslbf
this_stream_id		7	uimbsf
reserved	'1'	1	bslbf
bundle_id		7	uimbsf
if (this_is_main_stream) {			
reserved	'1'	1	bslbf
num_auxiliary_streams		7	uimbsf
for (m=0; m<num_auxiliary_streams; m++) {			
reserved	'1'	1	bslbf
auxiliary_stream_id		7	uimbsf
}			
}			
}			

this_is_main_stream equal 1 indicates that this stream contains a main stream that may be presented on its own, or that may be combined with additional audio components from an auxiliary stream. The main stream shall be delivered as an MMTP/MPU stream. Auxiliary streams are delivered as MMTP/MPU streams using different asset_IDs and they are signalled within the VR3DAudioAssetInformation message.

this_stream_id indicates the ID of this audio stream. This ID shall be unique within one bundle, i.e., for all streams that have the same bundle_id.

bundle_id identifies a unique ID for one bundle of audio streams. A bundle consists of exactly one main stream and one or more additional auxiliary streams that shall have the same bundle_id. The auxiliary streams contain additional audio components that may be combined with the main stream.

num_auxiliary_streams indicates the number of auxiliary streams that are available to be combined with the main stream.

auxiliary_stream_id identifies the ID of the auxiliary stream. The ID of all auxiliary streams shall be unique within one bundle.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

10 Media profiles

10.1 Video profiles

10.1.1 Overview

Subclause 10.1 defines media profiles for video. Table 31 provides an informative overview of the supported features. The detailed, normative specification for each video profile is subsequently provided in the referenced clause.

Table 31 – Overview of OMAF media profiles for video

Media profile	Codec	Profile	Level	Required scheme types	Brand	Clause
HEVC-based viewport-independent OMAF video profile	HEVC	Main 10	5.1	`podv` and `erpv`	`hevi`	10.1.2
HEVC-based viewport-dependent OMAF video profile	HEVC	Main 10	5.1	`podv` and at least one of `erpv` and `ercm`	`hevd`	10.1.3
AVC-based viewport-dependent OMAF video profile	AVC	Progressive High	5.1	`podv` and at least one of `erpv` and `ercm`	`avde`	10.1.4

NOTE For the HEVC Main 10 profile, the bit depth of decoded pictures could be either 8 bits or 10 bits.

10.1.2 HEVC-based viewport-independent OMAF video profile

10.1.2.1 General

Both monoscopic and stereoscopic spherical video up to 360 degrees are supported. The profile requires neither viewport-dependent delivery nor viewport-dependent decoding. Regular HEVC encoders, DASH packagers, DASH clients, file format parsers, and HEVC decoder engines that do not need special features for handling of viewport-dependent delivery and decoding could be used for encoding, distribution and decoding. The profile also minimizes the options for basic interoperability.

10.1.2.2 Elementary stream constraints

The elementary stream constraints apply to the HEVC bitstream that is reconstructed from a file as specified in subclause 10.1.2.5.

The bitstream shall comply with HEVC Main 10 profile, Main tier, Level 5.1.

All pictures shall be encoded as coded frames, and shall not be encoded as coded fields.

All the active SPSs of the bitstream shall be constrained as follows:

- general_progressive_source_flag shall be equal to 1.
- general_frame_only_constraint_flag shall be equal to 1.
- general_interlaced_source_flag shall be equal to 0.

When VUI is present, aspect_ratio_idc should not be present or aspect_ratio_idc should be equal to 0 (unspecified) or 1 (square).

NOTE When aspect_ratio_idc is not present, ISO/IEC 23008-2 specifies that aspect_ratio_idc is inferred to be equal to 0.

For each picture, there shall be an equirectangular projection SEI message present in the bitstream that applies to the picture.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

When present, a frame packing arrangement SEI message shall be constrained in either of the following ways:

- frame_packing_arrangement_cancel_flag shall be equal to 1.
- All of the following constraints apply:
  - frame_packing_arrangement_type shall be equal to 3, 4, or 5.
  - quincunx_sampling_flag shall be equal to 0.
  - content_interpretation_type shall be equal to 1.

When the video does not cover the entire sphere, for each picture, there shall be a region-wise packing SEI message present in the bitstream that applies to the picture.

When present, the region-wise packing SEI messages shall indicate constraints that comply with the equirectangular projected video scheme type `'erpv'` specified in subclause 7.6.1.3.

10.1.2.3 SEI message related ISO base media file format constraints

This subclause specifies ISOBMFF constraints depending on the presence of omnidirectional video SEI messages of HEVC. This subclause is applicable to both the HEVC-based viewport-independent OMAF video profile and the HEVC-based viewport-dependent OMAF video profile as specified subsequently.

The following constraints apply for each picture in the bitstream:

- When the bitstream contains an equirectangular projection SEI message applying to the picture, `ProjectionFormatBox` with `projection_type` equal to 0 shall be present in the sample entry applying to the sample containing the picture.
- When the bitstream contains a cubemap projection SEI message applying to the picture, `ProjectionFormatBox` with `projection_type` equal to 1 shall be present in the sample entry applying to the sample containing the picture.
- When the bitstream contains an equirectangular projection SEI message with `erp_padding_flag` equal to 1 applying to the picture, `RegionWisePackingBox` shall be present in the sample entry applying to the sample containing the picture and shall signal the same information as indicated with the values of `gb_erp_type`, `left_gb_erp_width`, and `right_gb_erp_width` syntax elements of the equirectangular projection SEI message, i.e., the following applies:
  - Let horFact be equal to `packed_picture_width` / `width`, where `width` is the syntax element of the `VisualSampleEntry` containing the `RegionWisePackingBox`, and let gbWidth be equal to `left_gb_erp_width` + `right_gb_erp_width`.
  - If SpatiallyPackedStereoFlag is equal to 1, `constituent_picture_matching_flag` shall be equal to 1. Otherwise, the value of `constituent_picture_matching_flag` shall be equal to 0.
  - The value of `proj_picture_height` shall be equal to `packed_picture_height`.
  - If SideBySideFlag is equal to 1, `proj_picture_width` shall be equal to `packed_picture_width` − horFact * 2 * gbWidth. Otherwise, `proj_picture_width` shall be equal to `packed_picture_width` − horFact * gbWidth.
  - `num_regions` shall be equal to 1.
  - `guard_band_flag[0]` shall be equal to 1.
  - `packing_type[0]` shall be equal equal to 0.
  - `proj_reg_width[0]` shall be equal equal to `proj_picture_width` / HorDiv1.
  - `proj_reg_height[0]` shall be equal equal to `proj_picture_height` / VerDiv1.
  - `proj_reg_top[0]` shall be equal to 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

- `proj_reg_left[0]` shall be equal to 0.
- `transform_type[0]` shall be equal to 0.
- `packed_reg_width[0]` shall be equal to `proj_picture_width` / HorDiv1.
- `packed_reg_height[0]` shall be equal to `proj_picture_height` / VerDiv1.
- `packed_reg_top[0]` shall be equal to 0.
- `packed_reg_left[0]` shall be equal equal to `left_gb_erp_width` * horFact.
- `left_gb_width[0]` shall be equal to `left_gb_erp_width` * horFact.
- `right_gb_width[0]` shall be equal to `right_gb_erp_width` * horFact.
- `top_gb_height[0]` shall be equal to 0.
- `bottom_gb_height[0]` shall be equal to 0.

NOTE When a file writer has no information whether sample values of the guard bands are used in the inter prediction process, `gb_not_used_for_pred_flag[0]` ought to be set equal to 0.

- `gb_type[0][j]` shall be equal to `gb_erp_type` for each value of j in the range of 0 to 3, inclusive.

- When the bitstream contains a frame packing arrangement SEI message applying to the picture, `StereoVideoBox` shall be present in the sample entry applying to the sample containing the picture. When `StereoVideoBox` is present, it shall signal the frame packing format that is included in the frame packing arrangement SEI message(s) in the elementary stream.
- When the bitstream contains a region-wise packing SEI message applying to the picture, `RegionWisePackingBox` shall be present in the sample entry applying to the sample containing the picture. When present, `RegionWisePackingBox` shall signal the same information as in the region-wise packing SEI message(s).

10.1.2.4 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'hevi'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'hevi'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'hevi'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

At least one sample entry type of each sample entry of the track shall be equal to `'resv'`.

NOTE 1 `'resv'` does not have to be the track sample entry type when the track has undergone several transformations. Consequently, this media profile could also be used when the track is protected.

The `scheme_type` values of `SchemeTypeBox` in the `RestrictedSchemeInfoBox` and of all instances of `CompatibleSchemeTypeBox` in the same `RestrictedSchemeInfoBox` shall include `'podv'` and `'erpv'`.

The untransformed sample entry type, as derived in subclause 7.1.7, shall be equal to `'hvc1'`.

NOTE 2 Consequently, parameter sets are not present inband within samples.

`LHEVCConfigurationBox` shall not be present in `VisualSampleEntry`.

`HEVCConfigurationBox` in `VisualSampleEntry` shall indicate conformance to the elementary stream constraints specified in subclause 10.1.2.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

The constraints specified in subclause 10.1.2.3 apply to the track conforming to this media profile, where the bitstream based on which the constraints are derived is reconstructed from the track as specified in subclause 10.1.2.5.

When the playback is intended to be started using another viewing orientation than that indicated by (`centre_azimuth`, `centre_elevation`, `centre_tilt`) equal to (0, 0, 0) relative to the global coordinate axes, the initial viewing orientation metadata, as specified in subclause 7.7.4, shall be present.

10.1.2.5 File decoding process

The inputs to the file decoding process are

- the `track_ID` value of the track conforming to this media profile, and
- a file containing at least the track.

An HEVC bitstream is reconstructed from the track with the given `track_ID` value as specified in ISO/IEC 14496-15:2017, Clause 8.

The HEVC bitstream shall conform to the elementary stream constraints specified in subclause 10.1.2.2.

The HEVC bitstream is decoded as specified in ISO/IEC 23008-2:2017, subclause 8.1.1. The outputs of this process are the same as the outputs of ISO/IEC 23008-2:2017, subclause 8.1.1. Additionally, for each decoded picture, this process outputs `ProjectionFormatBox`, `StereoVideoBox` (when applicable), `RegionWisePackingBox` (when applicable), and `RotationBox` (when applicable) that provide the input for the semantics of sample locations within the decoded picture as specified in subclause 7.5.1.

10.1.2.6 Expected OMAF player operation

OMAF players conforming to this media profile are expected to process either all referenced SEI messages in clause 10.1.2.2 or all allowed boxes within the `SchemeInformationBox` for the equirectangular projected video scheme type.

When playing a file and when the file contains the initial viewing orientation metadata, as specified in subclause 7.7.4, OMAF players are expected to parse the initial viewing orientation metadata track associated with a media track and obey it when rendering the media track.

When receiving and playing a DASH presentation and a Representation containing initial viewing orientation metadata, as specified in subclause 7.7.4, is present and associated with a media Representation selected for playing, as specified in subclause 8.2.3, OMAF players are expected to receive and parse the initial viewing orientation metadata Representation and obey it when rendering the media Representation.

10.1.3 HEVC-based viewport-dependent OMAF video profile

10.1.3.1 General

This profile allows unconstrained use of rectangular region-wise packing. With the presence of region-wise packing, the resolution or quality of the omnidirectional video could be emphasized in certain regions, e.g., according to the user's viewing orientation. In addition, the untransformed sample entry type `'hvc2'` is allowed, making it possible to use extractors and get a conforming HEVC bitstream when tile-based streaming is used.

10.1.3.2 Elementary stream constraints

The elementary stream constraints apply to the HEVC bitstream that is reconstructed from a file as specified in subclause 10.1.3.4.

The HEVC bitstream shall comply with the same constraints as the HEVC-based viewport-independent OMAF video profile, with the following exceptions:

- For each picture, there shall be either an equirectangular projection SEI message or a cubemap projection SEI message present in the bitstream that applies to the picture.
- When present, the region-wise packing SEI messages shall indicate constraints that comply with the equirectangular projected video scheme type `'erpv'` specified in subclause 7.6.1.3 or the packed equirectangular or cubemap projected video scheme type `'ercm'` specified in subclause 7.6.1.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

10.1.3.3 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'hevd'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'hevd'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'hevd'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

At least one sample entry type of each sample entry of the track shall be equal to `'resv'`.

NOTE 1 `'resv'` does not have to be the track sample entry type, when the track has undergone several transformations. Consequently, this media profile could also be used when the track is protected.

The `scheme_type` values of `SchemeTypeBox` in the `RestrictedSchemeInfoBox` and of all instances of `CompatibleSchemeTypeBox` in the same `RestrictedSchemeInfoBox` shall include `'podv'` and at least one of `'erpv'` and `'ercm'`.

The untransformed sample entry type, as derived in subclause 7.1.7, shall be equal to `'hvc1'` or `'hvc2'`.

When the untransformed sample entry type is `'hvc2'`, the track shall include one or more `'scal'` track references. The referred tracks shall conform to the HEVC-based viewport-independent OMAF video profile or the HEVC-based viewport-dependent OMAF video profile.

For each sub-picture bitstream carried in a referred track, the value of each sample in each decoded sub-picture is identical to the value of the corresponding sample in the decoded reconstructed picture.

NOTE 2 The following two constraints for each sub-picture bitstream carried in a referred track are sufficient to ensure the above constraint to be satisfied, but in some cases, they are more than necessary:

1) It is required that there are no sample values outside the picture that are referenced for inter prediction.

2) For prediction units located on the right-side picture boundary except the last one at the bottom-right corner of the picture, the following applies when CuPredMode[xPb][yPb] is equal to MODE_INTER, where (xPb, yPb) specifies the top-left sample of the corresponding luma prediction block relative to the top-left sample of the current picture:

   - With the number of spatial merging candidates numSpatialMergeCand derived as follows:

     $$numSpatialMergeCand = availableFlagA_0 + availableFlagA_1 + availableFlagB_0 + availableFlagB_1 + availableFlagB_2$$

     where $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$, and $availableFlagB_2$ are the output of the derivation process for spatial merging candidates specified in ISO/IEC 23008-2:2018, subclause 8.5.3.2.3, the following applies:

     - If numSpatialMergeCand is equal to 0, merge_flag[xPb][yPb] is required to be equal to 0.
     - Otherwise (numSpatialMergeCand is greater than 0), merge_idx[xPb][yPb] is required to be in the range of 0 to numSpatialMergeCand − 1, inclusive.

   - With the number of spatial motion vector predictor candidates numSpatialMvpCand derived as follows:

     ```
     if ( availableFlagLXA )
             numSpatialMvpCand = availableFlagLXA +
                                 ( ( mvLXA  !=  mvLXB ) ? availableFlagLXB : 0 )
     else
             numSpatialMvpCand = availableFlagLXB
     ```

     where availableFlagLXA, availableFlagLXB, mvLXA, and mvLXB are the output of the derivation process for motion vector predictor candidates from neighbouring prediction unit partitions specified in ISO/IEC 23008-2:2018, subclause 8.5.3.2.7, the following applies:

     - If numSpatialMvpCand is equal to 0, mvp_l0_flag[xPb][yPb] and mvp_l1_flag[xPb][yPb] are both required to be equal to 1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

– Otherwise (numSpatialMvpCand is greater than 0), mvp_l0_flag[xPb][yPb] and mvp_l1_flag[xPb][yPb] are both required to be in the range of 0 to numSpatialMvpCand − 1, inclusive.

Note that the first constraint above restricts that motion vectors point to full-sample locations inside the picture and to fractional-sample locations that require only full-sample locations inside the picture for interpolation. Note that in the above constraints and the previous sentence, a sub-picture becomes a picture within the context of one sub-picture bitstream. The second constraint restricts that, when the sub-picture bitstream of the referred track together with other sub-picture bitstreams carried in other referred tracks are reconstructed into one conforming bitstream, in decoding of the entire reconstructed bitstream, for blocks of the sub-picture of this sub-picture bitstream, there won't be motion vector candidates for temporal motion vector prediction derived from blocks outside the "sub-picture".

NOTE 3 Besides the above constraint, the following constraints are also expected to be satisfied:

1) When multiple tiles are present in the bitstream resolved from an extractor track, the value of entropy_coding_sync_enabled_flag in the active PPS is required be equal to 0 for each sub-picture bitstream carried in a referred track.

2) The referred tracks are required to contain the same number of media samples.

3) The samples with the same sample number across the referred tracks are required to have the same presentation time.

The pictures carried in the samples with the same sample number across the referred tracks are required to have the same value of picture order count, i.e., PicOrderCntVal.

`LHEVCConfigurationBox` shall not be present in `VisualSampleEntry`.

`HEVCConfigurationBox` in `VisualSampleEntry` shall indicate conformance to the elementary stream constraints specified in subclause 10.1.3.2.

The `track_not_intended_for_presentation_alone` flag of the `TrackHeaderBox` may be used to indicate that a track is not intended to be presented alone.

The constraints specified in subclause 10.1.2.3 apply to the track conforming to this media profile, where the bitstream based on which the constraints are derived is reconstructed from the track as specified in subclause 10.1.3.4.

When the playback is intended to be started using another viewing orientation than that indicated by (`centre_azimuth`, `centre_elevation`, `centre_tilt`) equal to (0, 0, 0) relative to the global coordinate axes, the initial viewing orientation metadata, as specified in subclause 7.7.4, shall be present.

10.1.3.4 File decoding process

The inputs to the file decoding process are

- the `track_ID` value of the track conforming to this media profile, and
- a file containing at least the track with that `track_ID` value and all tracks that the track directly or indirectly depends on.

An HEVC bitstream is reconstructed as follows:

- If the untransformed sample entry type of the track with the given `track_ID` is equal to `'hvc1'`, an HEVC bitstream is generated from the track with the given `track_ID` value as specified in ISO/IEC 14496-15:2017, Clause 8.
- Otherwise (the untransformed sample entry type of the track with the given `track_ID` is equal to `'hvc2'`), an HEVC bitstream is generated from the track with the given `track_ID` value as specified in ISO/IEC 14496-15:2017, Clause 9 and Annex A.

The HEVC bitstream shall conform to the elementary stream constraints specified in subclause 10.1.3.2.

The HEVC bitstream is decoded as specified in ISO/IEC 23008-2:2017, subclause 8.1.1. The outputs of this process are the same as the outputs of ISO/IEC 23008-2:2018, subclause 8.1.1. Additionally, for each decoded picture, this process outputs `ProjectionFormatBox`, `StereoVideoBox` (when applicable), `RegionWisePackingBox` (when applicable), and `RotationBox` (when applicable) that provide the input for the semantics of sample locations within the decoded picture as specified in subclause 7.5.1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

10.1.3.5 Expected OMAF player operation

OMAF players conforming to this media profile are expected to process either all referenced SEI messages in subclause 10.1.3.2 or all allowed boxes within the `SchemeInformationBox` for the `'erpv'` and `'ercm'` scheme types.

When playing a file and when the file contains the initial viewing orientation metadata, as specified in subclause 7.7.4, OMAF players are expected to parse the initial viewing orientation metadata track associated with a media track and obey it when rendering the media track.

When receiving and playing a DASH presentation and a Representation containing initial viewing orientation metadata, as specified in subclause 7.7.4, is present and associated with a media Representation selected for playing, as specified in subclause 8.2.3, OMAF players are expected to receive and parse the initial viewing orientation metadata Representation and obey it when rendering the media Representation.

A player conforming to the HEVC-based viewport-dependent OMAF video profile is expected to:

- Parse both `SphereRegionQualityRankingBox` and `2DRegionQualityRankingBox`, when present, of tracks present in a file and select the track that matches user's viewing orientation. The player should use `2DRegionQualityRankingBox` together with `RegionWisePackingBox`, when present, to conclude which sphere regions the indicated quality ranking 2D regions correspond to.
- Parse spherical region-wise quality ranking (SRQR) descriptors, when present, and select the Adaptation Sets and Representations that match the user's viewing orientation:
  - When Preselections are applied in the MPD, the player is expected to select a main Adaptation Set based on the SRQR descriptors.
  - When `@dependencyId` is applied in the MPD, the player is expected to select the dependent Representation based on the SRQR descriptors.

10.1.4 AVC-based viewport-dependent OMAF video profile

10.1.4.1 General

This media profile allows unconstrained use of rectangular region-wise packing with AVC. With the presence of region-wise packing, the resolution of the omnidirectional video could be emphasized in certain regions, e.g., according to the user's viewing orientation. In addition, the untransformed sample entry types `'avc2'` and `'avc4'` are allowed, making it possible to use extractors and get a conforming AVC bitstream when slice-based streaming is used.

10.1.4.2 Elementary stream constraints

The elementary stream constraints apply to the AVC bitstream that is reconstructed from a file as specified in subclause 10.1.4.4.

The bitstream shall comply with AVC progressive high profile, level 5.1.

10.1.4.3 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'avde'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'avde'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'avde'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

At least one sample entry type of each sample entry of the track shall be equal to `'resv'`.

NOTE `'resv'` does not have to be the track sample entry type, when the track has undergone several transformations. Consequently, this media profile could also be used when the track is protected.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

The scheme_type values of SchemeTypeBox in the RestrictedSchemeInfoBox and of all instances of CompatibleSchemeTypeBox in the same RestrictedSchemeInfoBox shall include 'podv' and at least one of 'erpv' and 'ercm'.

The untransformed sample entry type, as derived in clause 7.1.7, shall be equal to 'avc1', 'avc2', 'avc3', or 'avc4'.

When the untransformed sample entry type is 'avc2' or 'avc4', the track shall include one or more 'scal' track references.

AVCConfigurationBox in VisualSampleEntry shall indicate conformance to the elementary stream constraints specified in subclause 10.1.4.2.

When the playback is intended to be started using another viewing orientation than that indicated by (centre_azimuth, centre_elevation, centre_tilt) equal to (0, 0, 0) relative to the global coordinate axes, the initial viewing orientation metadata, as specified in subclause 7.7.4, shall be present.

10.1.4.4 File decoding process

The inputs to the file decoding process are

- the track_ID value of the track conforming to this media profile, and
- a file containing at least the track with that track_ID value and all tracks that the track directly or indirectly depends on.

An AVC bitstream is reconstructed as follows:

- If the untransformed sample entry type of the track with the given track_ID is equal to 'avc1' or 'avc3', an AVC bitstream is generated from the track with the given track_ID value as specified in ISO/IEC 14496-15:2017, Clause 5.
- Otherwise (the untransformed sample entry type of the track with the given track_ID is equal to 'avc2' or 'avc4'), an AVC bitstream is generated from the track with the given track_ID value as specified in ISO/IEC 14496-15:2017, Clause 5 and Annex A.

The AVC bitstream shall conform to the elementary stream constraints specified in subclause 10.1.4.2.

The AVC bitstream is decoded as specified in ISO/IEC 14496-10:2014, Clause 8. The outputs of this process are the same as the outputs of ISO/IEC 14496-10:2014, Clause 8. Additionally, for each decoded picture, this process outputs ProjectionFormatBox, StereoVideoBox (when applicable), RegionWisePackingBox (when applicable), and RotationBox (when applicable) that provide the input for the semantics of sample locations within the decoded picture as specified in subclause 7.5.1.

10.1.4.5 Expected OMAF player operation

OMAF players conforming to this media profile are expected to process all allowed boxes within the SchemeInformationBox for the 'erpv' and 'ercm' scheme types.

When playing a file and when the file contains the initial viewing orientation metadata, as specified in subclause 7.7.4, OMAF players are expected to parse the initial viewing orientation metadata track associated with a media track and obey it when rendering the media track.

When receiving and playing a DASH presentation and a Representation containing initial viewing orientation metadata, as specified in subclause 7.7.4, is present and associated with a media Representation selected for playing, as specified in subclause 8.2.3, OMAF players are expected to receive and parse the initial viewing orientation metadata Representation and obey it when rendering the media Representation.

A player conforming to the AVC-based viewport-dependent OMAF video profile is expected to:

- Parse both SphereRegionQualityRankingBox and 2DRegionQualityRankingBox, when present, of tracks present in a file and select the track that matches user's viewing orientation. The player should use 2DRegionQualityRankingBox together with RegionWisePackingBox, when present, to conclude which sphere regions the indicated quality ranking 2D regions correspond to.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

- Parse spherical region-wise quality ranking (SRQR) descriptors, when present, and select the Adaptation Sets and Representations that match the user's viewing orientation:
  - When Preselections are applied in the MPD, the player is expected to select a main Adaptation Set based on the SRQR descriptors.
  - When `@dependencyId` is applied in the MPD, the player is expected to select the dependent Representation based on the SRQR descriptors.

10.2 Audio profiles

10.2.1 Overview

Subclause 10.2 defines media profiles for audio in OMAF. Table 32 provides an informative overview of the supported features. The detailed, normative specification for each audio profile is subsequently provided in the referenced clause.

Table 32 - Overview of OMAF media profiles for audio

Media profile	Codec	Profile	Level	Max sampling rate	3D metadata	Brand	Clause
OMAF 3D audio baseline profile	MPEG-H Audio	Low Complexity	1, 2 or 3	48 kHz	included in codec	`oabl`	10.2.2
OMAF 2D audio legacy profile	AAC	HE-AACv2	4	48 kHz	no 3D metadata	`oa2d`	10.2.3

NOTE The audio streams complying with the MPEG-H 3D audio low complexity (LC) profile, levels 1 or 2 (i.e., `mpegh3daProfileLevelIndication` is set to "0x0B" or "0x0C", comply also with MPEG-H 3D audio LC profile, level 3 (i.e., `mpegh3daProfileLevelIndication` is set to "0x0D"), as specified in ISO/IEC 23008-3:2015, subclause 5.3.2.

10.2.2 OMAF 3D audio baseline profile

10.2.2.1 General

This media profile fulfills the requirements to support 3D audio. Channels, objects and higher-order ambisonics (HOA) are supported, as well as combinations of those. The profile is based on ISO/IEC 23008-3.

MPEG-H 3D audio (ISO/IEC 23008-3) specifies coding of immersive audio material and the storage of the coded representation in an ISOBMFF track. The MPEG-H 3D Audio decoder has a constant latency, see ISO/IEC 23008-3:2015, Table 1. With this information, content authors could synchronize audio and video portions of a media presentation, e.g. ensuring lip-synch. When orientation sensor inputs (i.e., azimuth, elevation) of an MPEG-H 3D audio decoder change, there will be some algorithmic and implementation latency (perhaps tens of ms) between user head movement and the desired sound field orientation. This latency will not impact audio/visual synchronization (i.e. lip synch), but only represents the lag of the rendered sound field with respect to the user head orientation.

MPEG-H 3D audio specifies methods for binauralizing the presentation of immersive content for playback via headphones, as is needed for omnidirectinal media presentations. MPEG-H 3D audio specifies a normative interface for the user's viewing orientation and permits low-complexity, low-latency rendering of the audio scene to any user orientation.

10.2.2.2 Elementary stream constraints

The audio stream shall comply with the MPEG-H 3D audio low complexity (LC) profile, levels 1, 2 or 3 as defined in ISO/IEC 23008-3:2015, subclause 4.8. The values of the `mpegh3daProfileLevelIndication` for LC profile levels 1, 2 and 3 are "0x0B", "0x0C" and "0x0D", respectively, as specified in ISO/IEC 23008-3:2015, subclause 5.3.2.

Audio data shall be encapsulated into MPEG-H audio stream (MHAS) packets according to ISO/IEC 23008-3:2015, Clause 14.

All MHAS packet types defined in ISO/IEC 23008-3:2015, Clause 14 may be present in the stream, except of the following packet types that shall not be present in the stream:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- PACTYP_CRC16
- PACTYP_CRC32
- PACTYP_GLOBAL_CRC16
- PACTYP_GLOBAL_CRC32

If audio scene information per ISO/IEC 23008-3:2015, Clause 15 is present, it always shall be encapsulated in an MHAS PACTYP_AUDIOSCENEINFO packet. audio scene information shall not be included in the mpegh3daConfig() structure in the MHAS PACTYP_MPEGH3DACFG packet.

10.2.2.3 ISO base media file format constraints

10.2.2.3.1 General constraints

When a track is the only track in a file, compatible_brands containing a brand equal to 'oabl' in FileTypeBox indicates that the track conforms to this media profile. When a file contains multiple tracks, compatible_brands containing a brand equal to 'oabl' in FileTypeBox indicates that at least one of the tracks conforms to this media profile.

compatible_brands containing a brand equal to 'oabl' in TrackTypeBox indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of FileTypeBox and TrackTypeBox.

The sample entry 'mhm1' shall be used for encapsulation of MHAS packets into ISOBMFF files, per ISO/IEC 23008-3:2015, subclause 20.6.

The sample entry 'mhm2' shall be used in cases of multi-stream delivery, i.e., the MPEG-H Audio Scene is split into two or more streams for delivery as described in ISO/IEC 23008-3:2015, subclause 14.6.

If the MHAConfigurationBox() is present, the MPEG-H profile and level indicator mpegh3daProfileLevelIndication in the MHADecoderConfigurationRecord() shall be set to "0x0B", "0x0C", or "0x0D" for MPEG-H Audio LC Profile Level 1, Level 2, or Level 3, respectively, as specified in ISO/IEC 23008-3:2015, subclause 5.3.2.

The first sample of the movie and the first sample of every fragment (when applicable) shall be a stream access point (SAP) of type 1 (i.e., sync sample). For MPEG-H Audio a sync sample shall be properly signalled according to ISO/IEC 14496-12. All rules defined in ISO/IEC 23008-3:2015, subclause 20.6.1 regarding sync samples shall apply. In, addition, a sync sample shall consist of MHAS packets in the following order:

- PACTYP_MPEGH3DACFG
- PACTYP_AUDIOSCENEINFO (if audio scene information is present)
- PACTYP_BUFFERINFO
- PACTYP_MPEGH3DAFRAME

Additional MHAS packets may be present between the MHAS packets listed above or after the MHAS packet PACTYP_MPEGH3DAFRAME, with one exception: if present, the PACTYP_AUDIOSCENEINFO packet shall directly follow the PACTYP_MPEGH3DACFG packet, as defined in ISO/IEC 23008-3:2015, subclause 14.4.

MPEG-H audio sync samples contain Immediate Playout Frames (IPFs), as specified in ISO/IEC 23008-3:2015, subclause 20.2, thus the audio data encapsulated in the MHAS packet PACTYP_MPEGH3DAFRAME shall contain the AudioPreRoll() syntax element, as defined in ISO/IEC 23008-3:2015, subclause 5.5.6, and shall follow the requirements for stream access points as defined in ISO/IEC 23008-3:2015, subclause 5.7. The audio configuration is delivered as part of the MHAS packet PACTYP_MPEGH3DACFG and, therefore, the AudioPreRoll() structure carried in the MHAS packet PACTYP_MPEGH3DAFRAME shall not contain the Config() structure, i.e., the configLen field of the AudioPreRoll() shall be equal to 0.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

10.2.2.3.2 Configuration change constraints

A configuration change takes place in an audio stream when the content setup or the audio scene information changes (e.g., when changes occur in the channel layout, the number of objects etc.), and therefore new `PACTYP_MPEGH3DACFG` and `PACTYP_AUDIOSCENEINFO` packets are required upon such occurrences. A configuration change usually happens at program boundaries, but it may also occur within a program.

The following constraints apply:

- At each configuration change, the `MHASPacketLabel` shall be changed to a different value from the `MHASPacketLabel` in use before the configuration change occurred. A configuration change may happen at the beginning of a new ISOBMFF file or at any position within the file. In the latter case, the file format sample that contains a configuration change shall be encoded as a sync sample (RAP) as defined above.
- A sync sample that contains a configuration change and the last sample before such a sync sample may contain a truncation message (i.e., a `PACTYP_AUDIOTRUNCATION` packet in the MHAS stream) as defined in ISO/IEC 23008-3:2015, subclause 14.4. If MHAS packets of type `PACTYP_AUDIOTRUNCATION` are present, they shall be used as described in ISO/IEC 23008-3:2015, subclause 14.4.

ISOBMFF tracks that belong to one audio programme use different configurations and a switch between two ISOBMFF tracks represents also a configuration change. Thus, the `MHASPacketLabel` needs to have different values for all ISOBMFF tracks that belong to one audio programme. Also, after a configuration change the `MHASPacketLabel` needs to have different values for all ISOBMFF tracks comprising an audio programme.

10.2.2.3.3 Multi-stream constraints

The multi-stream-enabled MPEG-H audio system is capable of handling audio programme components delivered in several different elementary streams (e.g., the main MHAS stream containing one complete audio main, and one or more auxiliary MHAS streams, containing different languages and audio description). The MPEG-H audio metadata information (MAE) allows the MPEG-H audio decoder to correctly decode several MHAS streams.

The following constraints apply for file formats using the sample entry `'mhm2'`:

- One MHAS stream shall be the main stream, i.e., in exactly one MHAS stream the audio scene information shall have the `mae_isMainStream` field set to 1. In all other MHAS streams the `mae_isMainStream` shall be set to 0.
- In each auxiliary MHAS stream (i.e., streams with `mae_isMainStream` field set to 0) the `mae_bsMetaDataElementIDoffset` field in the audio scene information shall be set to the index of the first metadata element in the auxiliary MHAS stream minus one.
- All MHAS elementary streams that carry audio programme components of one audio programme shall be time aligned.
- In each auxiliary MHAS elementary stream (i.e., streams with `mae_isMainStream` field set to 0), RAPs shall be aligned to the RAPs present in the main stream (i.e., the stream with `mae_isMainStream` field set to 1).
- Presentation Description Manifests need to make sure that all streams that contribute to one audio programme may be identified as such.
- For the main and the auxiliary MHAS stream(s), the `MHASPacketLabel` shall be set according to ISO/IEC 23008-3:2015, subclause 14.6. ISOBMFF tracks that belong to one switching set need to use different `MHASPacketLabel` values within the same range of values associated to one stream, as specified in ISO/IEC 23008-3:2015, subclause 14.6. For example, all ISOBMFF tracks in the switching set for the main stream use different values between 1 and 16, all ISOBMFF tracks in the switching set for the first auxiliary stream use values between 17 and 32, and so on.

10.2.2.3.4 Loudness and dynamic range control

Loudness metadata shall be embedded within the `mpegh3daLoudnessInfoSet()` structure as defined in ISO/IEC 23008-3:2015, subclause 6.3. Such loudness metadata shall include at least the loudness of the content rendered to the default rendering layout as indicated by the `referenceLayout` field (see ISO/IEC 23008-3:2015, subclause 5.3.2). More precisely, the `mpegh3daLoudnessInfoSet()` structure shall include at least one `loudnessInfo()` structure with `loudnessInfoType` set to 0, whose `drcSetId` and `downmixId` fields are set to 0 and which includes at least one `methodValue` field with `methodDefinition` set to 1 or 2 (see ISO/IEC 23008-3:2015, subclause 6.3.1 and

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

ISO/IEC 23003-4:2015, subclause 7.3). The indicated loudness value shall be measured according to applicable regional loudness regulations.

DRC metadata shall be embedded in the `mpegh3daUniDrcConfig()` and `uniDrcGain()` structures as defined in ISO/IEC 23008-3:2015, subclause 6.3. For each included DRC set the `drcSetTargetLoudnessPresent` field as defined in ISO/IEC 23003-4:2015, Clause 7 shall be set to 1.

The `bsDrcSetTargetLoudnessValueUpper` and `bsDrcSetTargetLoudnessValueLower` fields shall be configured to continuously cover the range of target loudness levels between -31 dB and 0 dB. The embedded DRC metadata should allow for a decoder output loudness of at least -16 LKFS.

Loudness compensation information (`mae_LoudnessCompensationData()`), as defined in ISO/IEC 23008-3:2015, subclause 15.5 shall be present in the audio scene information if the `mae_allowGainInteractivity` field (according to ISO/IEC 23008-3:2015, subclause 15.3) is set to 1 for at least one group of audio elements.

10.2.3 OMAF 2D audio legacy profile

10.2.3.1 General

This media profile fulfills requirements to support 2D channel-based audio. The delivery of up to 5.1 channels is supported. The profile is based on MPEG-4 AAC specified in ISO/IEC 14496-3, which defines coding of general audio content. The delivery of up to 5.1 audio channels allows 2D rendering according to the user's viewing orientation.

HE-AAC is used worldwide in the most successful streaming services and supported by all major streaming and media platforms. Due to the wide reach, MPEG-4 AAC may be used for VR services and platforms, which use either mono, stereo, 4.0, or 5.1 surround channel configurations. The 2D audio legacy profile does not require any new signalling for the audio codec and its configuration. Therefore, it is compatible with all decoder implementations in the market.

10.2.3.2 Elementary stream constraints

10.2.3.2.1 General encoding constraints

The audio stream shall comply with MPEG-4 AAC-LC, HE-AAC or HE-AACv2 profiles, level 4, as defined in ISO/IEC 14496-3.

For HE-AAC encoded tracks, the first sample of the ISOBMFF movie and the first sample of every ISOBMFF movie fragment (when applicable) shall be a SAP of type 1, notably, the SBR configuration information shall be present in the audio access unit.

ISOBMFF tracks containing AAC audio as defined in ISO/IEC 14496-3 shall conform to the following AAC audio encoding constraints:

- The elementary stream shall be a raw data stream, i.e., ADTS and ADIF headers shall not be present.
- Each AAC elementary stream shall be encoded using MPEG-4 AAC LC, HE-AAC, HE-AACv2, Level 4. Use of the MPEG-4 HE-AACv2 for stereo configuration is recommended for 32 kbps or lower.
- When using HE-AAC and HE-AACv2, explicit backwards compatible signalling shall be used to indicate the use of the SBR and PS coding tools.
- AAC elementary streams shall not exceed 48kHz sampling rate.
- The number of channels, including the LFE channel, of an AAC ISOBMFF track shall not exceed six audio channels.
- AAC ISOBMFF fragments containing HE-AAC shall start with a type 1 SAP, notably, the SBR configuration information shall be in the first packet.
- The transform length of the IMDCT for AAC shall be 1024 audio PCM samples for long blocks, and 128 audio PCM samples for short blocks.
- The following parameters shall not change within the elementary stream
  - audio object type
  - sampling frequency

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

– channel configuration

The `channelConfiguration` parameter carried in the `AudioSpecificConfig` shall be set according to one of the following specified values:

– `channelConfiguration` is set equal to 1 for mono audio.

– `channelConfiguration` is set equal to 2 for stereo audio.

– `channelConfiguration` is set equal to 4 for four channel audio.

– `channelConfiguration` is set equal to 5 for five channel audio.

– `channelConfiguration` is set equal to 6 for six channel audio, i.e., 5.1 audio.

Producing audio content capable of seamless bitrate adaptation with OMAF 2D audio legacy media profile (AAC-LC, HE-AAC, HE-AACv2) requires constrained encoding at fragment boundaries. For such scenarios, each AAC elementary stream shall be encoded following the constraints provided in ISO/IEC 23000-19:2018, subclauses 10.5.2 to 10.5.6.

Encoding recommendations for AAC audio tracks are provided in ISO/IEC 23000-19:2018, Annex G.

10.2.3.2.2 Syntax and values of syntactic elements

The syntax and values for syntactic elements shall conform to ISO/IEC 14496-3. The following element shall not be present in an MPEG-4 HE-AAC or HE-AACv2 elementary stream:

– coupling_channel_element (CCE)

If the program_config_element (PCE) element is present then it shall only list a set of channels corresponding to one of the fixed channel configurations specific in ISO/IEC 14496-3:2009, Table 1.19, and the element shall not change for the duration of the track.

The arrangement of syntactic elements shall be according to ISO/IEC 14496-3:2009. Table 1.19. For convenience, the arrangement of elements for the allowed channel configurations is reported in Table 33.

Table 33 – Arrangement of Audio syntactic elements

Channel configuration	Number of channels	Audio syntactic elements
1	1	<SCE>, <optional additional elements>, <TERM>, for HE-AAC v2, and mono HE-AAC or AAC-LC
2	2	<CPE>, <optional additional elements>, <TERM>, for stereo HE-AAC or AAC-LC
4	4	<SCE>, <CPE>, <SCE>, <optional additional elements>, <TERM>
5	5.0	<SCE>, <CPE>, <CPE>, <optional additional elements>, <TERM>
6	5.1	<SCE>, <CPE>, <CPE>, <LFE>, <optional additional elements>, <TERM>

NOTE Angled brackets (<>) are used above to indicate separate syntactic elements, not stream syntax.

The syntax and values for `individual_channel_stream` shall conform to ISO/IEC 14496-3. The following fields shall be set as follows:

– `gain_control_data_present` is set equal to 0.

10.2.3.2.3 AAC presentation timing

The AAC codec uses audio frames of a fixed length, and a transform which applies over two frames. To obtain correct audio from a frame, both frames in the transform are needed, and hence the prior encoded frame and the current encoded frame need to be decoded to output the first frame. This is sometimes called "priming" and may be signalled using the `'roll'` sample group.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

A full reconstruction of the first encoded audio frame is sometimes not possible since there is no previous access unit. To still achieve a full reconstruction, a common practice is to add silence to the beginning of the audio signal. A more detailed explanation of this approach may be found in ISO/IEC 14496-24.

In practice, an encoder might prepend an arbitrary amount of (invalid) audio waveform samples to the signal. This portion of the audio signal is sometimes called "encoder delay" and varies depending on the implementation.

Presentation delay is compensated according to one of the following options:

– The most common approach to compensate for inserted extra audio is to add an offset edit list to the ISOBMFF header. In the case where padding has been added to the start of an audio stream, the `media_time` in the edit list is the length (in audio samples, as measured by the timescale of the track) of the inserted audio samples; 2112 is a common example for AAC.

– If the content has been generated according to ISO/IEC 23000-19:2018, Annex G.5, no `EditListBox` is present.

– If the SBR and PS coding tools are present, they shall not be considered for the purpose of delay compensation.

10.2.3.2.4 Loudness and dynamic range control

The audio stream should contain DRC and loudness metadata according to ISO/IEC 14496-3. The audio encoder should set the Program Reference Level to the loudness level of the audio stream.

The audio encoder should generate DRC metadata for light compression encoded in the `dyn_rng_ctl` and `dyn_rng_sgn` fields of `dynamic_range_info()` in the FIL element and DRC metadata for heavy compression in the `compression_value` field of `MPEG4_ancillary_data()` in the data stream element (DSE).

NOTE It is expected that the audio decoder will use the Program Reference Level, if available, to achieve a desired target loudness, if applicable. It is expected that the audio decoder will apply the DRC metadata, if present, according to ISO/IEC 14496-3 including the DRC Presentation Mode value of the `drc_presentation_mode` fields.

10.2.3.2.5 Maximum bitrate

The maximum bitrate of AAC elementary streams shall be calculated in accordance with the AAC buffer requirements as defined in ISO/IEC 14496-3:2009, subclause 4.5.3. Only the raw data stream shall be considered in determining the maximum bitrate (system-layer descriptors are excluded).

10.2.3.3 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'oa2d'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'oa2d'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'oa2d'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

The syntax and values of the `AudioSampleEntry` shall conform to `MP4AudioSampleEntry` (`'mp4a'`) as defined in ISO/IEC 14496-14. Table 34 lists the allowed AAC profiles.

Table 34 – AAC profiles

AAC profile	codingname	Sample entry
MPEG-4 AAC (AAC-LC)	mp4a	`MP4AudioSampleEntry`
MPEG-4 High Efficiency AAC (HE-AAC)	mp4a	`MP4AudioSampleEntry`
MPEG-4 High Efficiency AAC v2 (HE-AACv2)	mp4a	`MP4AudioSampleEntry`

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

The SampleEntry format in the SampleDescriptionBox is the same for each AAC audio profile.

10.2.3.3.1 Storage of AAC media samples

The following additional constraints apply:

- All audio media samples shall consist of one AAC audio access unit.
- All AAC access units in an ISOBMFF track shall be encoded with one of AAC LC, HE-AAC or HE-AACv2.
- The values given in AudioSampleEntry, DecoderConfigDescriptor, and DecoderSpecificInfo shall match the corresponding values in the AAC audio bitstream.

10.2.3.3.2 AAC audio sample entry

The syntax and values of the AudioSampleEntry shall conform to MP4AudioSampleEntry ('mp4a') as defined in ISO/IEC 14496-14.

The sample entry and fields specified in this section shall not change within an ISOBMFF track.

The value of the channelcount parameter in the AudioSampleEntry box defined in ISO/IEC 14496-3 shall be set to one of the following specified values:

- channelcount is set equal to 1 for mono audio.
- channelcount is set equal to 2 for stereo audio.
- channelcount is set equal to 4 for four channel audio.
- channelcount is set equal to 5 for five channel audio.
- channelcount is set equal to 6 for six channel audio, i.e., 5.1 audio.

The value of the channelcount parameter in the AudioSampleEntry box shall correspond to the values of channelConfiguration field of AudioSpecificConfig according to Table 35:

Table 35 - Mapping of channelcount parameter in the AudioSampleEntry to channelConfiguration field of AudioSpecificConfig

channelcount	channelConfiguration
1	1
2	2
4	4
5	5
6	6
NOTE The channel to loudspeaker mapping for each channelConfiguration index is given in ISO/IEC 14496-3:2009, Table 1.19. The informative geometric speaker positions for channelConfiguration = 4 (Quadrophonic speaker layout) is 0, 90°, -90°, 180° deg (azimuth)	

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

10.2.3.3.2.1 ES_Descriptor

The syntax and values for ES_Descriptor shall conform to ISO/IEC 14496-1, and the fields of the ES_Descriptor shall be set as follows:

- ES_ID is set equal to 0.
- streamDependenceFlag is set equal to 0.
- URL_Flag is set equal to 0.
- OCRstreamFlag is set equal to 0.

© ISO/IEC 2019 – All rights reserved

- `streamPriority` is set equal to 0.
- `decConfigDescr` is set equal to `DecoderConfigDescriptor`.
- `slConfigDescr` is set equal to `SLConfigDescriptor`, predefined type 2.

Descriptors other than those specified in subclauses 10.2.3.3.2.2 through 10.2.3.3.2.4 shall not be used.

10.2.3.3.2.2 DecoderConfigDescriptor

The syntax and values for `DecoderConfigDescriptor` shall conform to ISO/IEC 14496-1, and the fields of this descriptor shall be constrained to the following values.

- `decoderSpecificInfo` shall be used, and `ProfileLevelIndicationIndexDescriptor` shall not be used.
- `objectTypeIndication` is equal to 0x40 (audio).
- `streamType` is equal to 0x05 (audio stream).
- `upStream` is equal to 0.
- `decSpecificInfo` is equal to `AudioSpecificConfig`.

10.2.3.3.2.3 AudioSpecificConfig

The syntax and values for `AudioSpecificConfig` shall conform to ISO/IEC 14496-3 .

The following fields of `AudioSpecificConfig` shall be set according to ISO/IEC 14496-3 and subclause 10.2.3.2:

- `audioObjectType`
- `channelConfiguration`
- `extensionAudioObjectType`
- `GASpecificConfig`

10.2.3.3.2.4 GASpecificConfig

The syntax and values for `GASpecificConfig` shall conform to ISO/IEC 14496-3, and the fields of `GASpecificConfig` shall be set to the following values:

- `frameLengthFlag` is equal to 0 (1024 lines IMDCT).
- `dependsOnCoreCoder` is equal to 0.
- `extensionFlag` is equal to 0.

10.3 Image profiles

10.3.1 Overview

Subclause 10.3 defines OMAF media profiles for image coding. Table 36 provides an informative overview of the supported features. The detailed, normative specification for each image profile is provided in the referenced clause. Common text for both profiles is provided in subclause 10.3.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Table 36 – Overview of OMAF media profiles for image coding

Media profile	Codec	Profile	Level	Brand	Clause
OMAF HEVC image profile	HEVC	Main 10	5.1	`heoi`	10.3.3
OMAF legacy image profile	JPEG	Not applicable	Not applicable	`jpoi`	10.3.4

10.3.2 Common specifications for image profiles

10.3.2.1 General

Subclause 10.3.2 specifies common constraints applicable to both OMAF image profiles. Both monoscopic and stereoscopic images up to 360 degrees are supported. These image profiles require the use of the equirectangular projection or the cubemap projection. When equirectangular projection and region-wise packing are in use, the number of regions in region-wise packing is required to be equal to 1 or 2 for monoscopic or stereoscopic content, respectively. When cubemap projection is in use, it is allowed to use up to one region per cube face in region-wise packing. The `'grid'` derived image item could be used to support large image widths and heights.

10.3.2.2 ISO base media file format constraints for an image item conforming to an OMAF image profile

An image item is specified to conform to an OMAF image profile when all of the following constraints are true:

- Either one of the following applies:
  - The image item is a coded image item conforming to the OMAF image profile.
  - The image item is a `'grid'` derived image item, and each source image item of the derived image item is a coded image item conforming to the OMAF image profile.
- The item properties for the coded image item comply with the requirements and constraints specified in subclause 7.9.
- The item is not associated with any other types of essential item properties than those required by the item type of the image item, and `'stvi'`, `'prfr'`, `'rwpk'`, and `'rotn'`, which are specified in this document.
- The item is associated with a `'prfr'` item property that indicates the equirectangular or cubemap projection.
- When the item is associated with a `'rwpk'` item property, the values of the item property are constrained as follows:
  - When equirectangular projection is in use, the value of NumRegions, derived as specified in subclause 7.5.1.2, shall be equal to HorDiv1 * VerDiv1, derived as specified in subclause 7.9.3.1.
  - When cubemap projection is in use, the number of projected regions containing samples of a particular cube face shall not be greater than 1.

NOTE When the image item covers less than 360 degrees, a cube face could be absent in a packed picture.

  - The value of PackingType[i] for each value of i in the range of 0 to NumRegions − 1, inclusive, derived as specified in subclause 7.5.1.2, shall be equal to 0.
- When the viewing of the image item is intended to be started using another viewing orientation than that indicated by (`centre_azimuth`, `centre_elevation`, `centre_tilt`) equal to (0, 0, 0) relative to the global coordinate axes, the initial viewing orientation item property, as specified in clause 7.9.9, shall be present and associated with the image item.

10.3.2.3 ISO base media file format constraints for a file conforming to an OMAF image profile

Each file including a four-character code of an OMAF image profile as a compatible brand shall conform to all of the following:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- The file shall include 'mif1' among the compatible brands and comply with the requirements of 'mif1' brand as specified in ISO/IEC 23008-12.
- The file shall contain at least one image item that conforms to all of the following:
  - The image item is present in the file.
  - When the image item is a derived image item, each source image item is present in the file.
  - The image item is either the primary item or any item from the alternate group containing the primary item.
  - The image item conforms to the OMAF image profile.

10.3.3 OMAF HEVC image profile

10.3.3.1 General

Subclause 10.3.3 specifies the OMAF HEVC image profile, which uses HEVC as the codec for coding of the image and follows the common constraints for OMAF image profiles.

10.3.3.2 Elementary stream constraints

The bitstream of a coded image item conforming to this media profile shall conform to HEVC Main 10 profile, Main tier, Level 5.1.

The bitstream contained in a coded image item shall consist of one and only one coded picture. The coded picture shall be a coded frame.

The active SPS of the bitstream shall be constrained as follows:

- general_progressive_source_flag shall be set to 1.
- general_frame_only_constraint_flag shall be set to 1.
- general_interlaced_source_flag shall be set to 0.

10.3.3.3 ISO base media file format constraints

A coded image item is specified to conform to the 'heoi' brand when all of the following constraints are true:

- The content of the coded image item conforms to the elementary stream constraints specified in subclause 10.3.3.2.
- The item has type 'hvc1' and conforms to the requirements imposed by the 'hvc1' item type, as specified in ISO/IEC 23008-12.
- The coded image item conforms to the constraints specified in subclause 10.3.2.2.

An image item is specified to conform to the 'heoi' brand when it is a coded image item conforming to the 'heoi' brand as specified above or a derived image item conforming to the constraints specified in subclause 10.3.2.2 and for which each source image item is a coded image item conforming to the 'heoi' brand.

10.3.3.4 File decoding process

The inputs to the file decoding process are

- the item_ID value of the image item conforming to this media profile, and
- a file containing at least the image item.

If the image item with the given item_id value is a coded image item, the following applies:

- An HEVC bitstream consists of the content of the item with the given item_ID value.
- The HEVC bitstream shall conform to the elementary stream constraints specified in subclause 10.3.3.2.
- The HEVC bitstream is decoded as specified in ISO/IEC 23008-2:2017, subclause 8.1.1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- The outputs of this process are the same as the outputs of ISO/IEC 23008-2:2018, subclause 8.1.1.

Otherwise (the image item is a `'grid'` derived image item), the following applies:

- For each source image item of the derived image item, the following applies:
  - An HEVC bitstream consists of the content of the source image item.
  - The HEVC bitstream shall conform to the elementary stream constraints specified in subclause 10.3.3.2.
  - The HEVC bitstream is decoded as specified in ISO/IEC 23008-2:2017, subclause 8.1.1.
- The output of this process is formed by tiling the reconstructed images resulting from the decoding of all the source image items, as specified in the image grid derivation of ISO/IEC 23008-12.

Additionally, this process outputs `ProjectionFormatProperty`, `FramePackingProperty` (when applicable), `RegionWisePackingProperty` (when applicable), and `RotationProperty` (when applicable) that provide the input for the semantics of sample locations within the decoded picture as specified in subclause 7.5.1.

10.3.3.5 Recommendations and requirements for OMAF player

The requirements on readers conforming to the `'mif1'` brand, as specified in ISO/IEC 23008-12, shall be supported.

Players conforming to the `'heoi'` brand shall support displaying an image item that conforms to both of the following:

- The image item is either the primary item or any item from the alternate group containing the primary item.
- The image item conforms to the `'heoi'` brand as specified in subclause 10.3.3.3.

When displaying an image item conforming to the `'heoi'` brand, players are expected to obey semantics of the sample locations of the decoded picture as specified in subclause 7.5.1.

When an image item conforming to the `'heoi'` brand is associated with the initial viewing orientation item property, a player is expected to parse the initial viewing orientation item property and obey it when viewing the item.

10.3.4 OMAF legacy image profile

10.3.4.1 General

Subclause 10.3.4 specifies OMAF legacy image profile, which uses JPEG as the codec for coding of the image and follows the common constraints for OMAF image profiles.

10.3.4.2 Elementary stream constraints

The elementary stream constraints are identical to those for the `'jpeg'` brand specified in ISO/IEC 23008-12.

10.3.4.3 ISO base media file format constraints

A coded image item is specified to conform to the `'jpoi'` brand when all of the following constraints are true:

- The content of the coded image item conforms to the elementary stream constraints specified in subclause 10.3.4.2.
- The item has type `'jpeg'` and conforms to the requirements imposed by the `'jpeg'` item type, as specified in ISO/IEC 23008-12, or is coded with MIME type `'image/jpeg'` and conforms to that MIME type specification.
- The coded image item conforms to the constraints specified in subclause 10.3.2.2.

An image item is specified to conform to the `'jpoi'` brand when it is a coded image item conforming to the `'jpoi'` brand as specified above or a derived image item conforming to the constraints specified in subclause 10.3.2.2 and for which each source image item is a coded image item conforming to the `'jpoi'` brand.

10.3.4.4 File decoding process

The inputs to the file decoding process are

- the `item_ID` value of the image item conforming to this media profile, and

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

- a file containing at least the image item.

If the image item with the given `item_id` value is a coded image item, the following applies:

- A JPEG bitstream consists of the content of the item with the given `item_ID` value.
- The JPEG bitstream shall conform to the elementary stream constraints specified in subclause 10.3.4.2.
- The JPEG bitstream is decoded as specified in ISO/IEC 10918-1.
- The output of this process is a decoded JPEG image.

Otherwise (the image item is a `'grid'` derived image item), the following applies:

- For each source image item of the derived image item, the following applies:
  - A JPEG bitstream consists of the content of the source image item.
  - The JPEG bitstream shall conform to the elementary stream constraints specified in subclause 10.3.4.2.
  - The JPEG bitstream is decoded as specified in ISO/IEC 10918-1.
- The output of this process is formed by tiling the reconstructed images resulting from the decoding of all the source image items, as specified in the image grid derivation of ISO/IEC 23008-12.

Additionally, this process outputs `ProjectionFormatProperty`, `FramePackingProperty` (when applicable), `RegionWisePackingProperty` (when applicable), and `RotationProperty` (when applicable) that provide the input for the semantics of sample locations within the decoded picture as specified in subclause 7.5.1.

10.3.4.5 Recommendations and requirements for OMAF player

The requirements on readers conforming to the `'mif1'` brand, as specified in ISO/IEC 23008-12, shall be supported.

Players conforming to the `'jpoi'` brand shall support displaying an image item that conforms to both of the following:

- The image item is either the primary item or any item from the alternate group containing the primary item.
- The image item conforms to the `'jpoi'` brand as specified in subclause 10.3.4.3.

When displaying an image item conforming to the `'jpoi'` brand, players are expected to obey semantics of the sample locations of the decoded picture as specified in subclause 7.5.1.

When an image item conforming to the `'jpoi'` brand is associated with the initial viewing orientation item property, a player is expected to parse the initial viewing orientation item property and obey it when viewing the item.

10.4 Timed text profiles

10.4.1 Overview

Subclause 10.4 defines media profiles for timed text. Timed text is used for providing subtitles and closed captions for omnidirectional video. Table 37 provides an informative overview of the supported features. The detailed, normative specification for each timed text profile is subsequently provided in the referenced clause.

Table 37 – Overview of OMAF media profiles for timed text

Media profile	Codec	Profile	Brand	Clause
OMAF IMSC1 timed text profile	IMSC1	Text Profile or Image Profile	`ttml`	10.4.2
OMAF WebVTT timed text profile	WebVTT	n\a	`ttwv`	10.4.3

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

10.4.2 OMAF IMSC1 timed text profile

10.4.2.1 Elementary stream constraints

The elementary stream shall conform to the text profile or image profile specified in W3C Recommendation, *TTML profiles for Internet media subtitles and captions 1.0 (IMSC1)*.

10.4.2.2 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'ttml'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'ttml'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'ttml'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

An IMSC1 track shall conforms to the IMSC1 track format as specified in subclause 7.10.3.

The media handler type is `'subt'`, and the track uses a subtitle media header.

The role of an IMSC1 track should be labelled by using the `KindBox`.

When timed text cues are displayed on sphere regions, the timed text sphere region metadata track, as specified in subclause 7.7.6, shall be present.

10.4.3 OMAF WebVTT timed text profile

10.4.3.1 Elementary stream constraints

The elementary stream shall conform to WebVTT: *The web video text tracks format*.

10.4.3.2 ISO base media file format constraints

When a track is the only track in a file, `compatible_brands` containing a brand equal to `'ttwv'` in `FileTypeBox` indicates that the track conforms to this media profile. When a file contains multiple tracks, `compatible_brands` containing a brand equal to `'ttwv'` in `FileTypeBox` indicates that at least one of the tracks conforms to this media profile.

`compatible_brands` containing a brand equal to `'ttwv'` in `TrackTypeBox` indicates that the track conforms to this media profile.

A track of this media profile shall be indicated to conform to this media profile through one or both of `FileTypeBox` and `TrackTypeBox`.

A WebVTT track shall conform to the WebVTT track format as specified in subclause 7.10.4, using a track `handler_type` of `'text'` with a `codingname` of `'wvtt'`.

The role of a WebVTT track should be labelled by using the `KindBox`.

When timed text cues are displayed on sphere regions, the timed text sphere region metadata track, as specified in subclause 7.7.6, shall be present.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

11 Presentation profiles

11.1 OMAF viewport-independent baseline presentation profile

11.1.1 General

The OMAF viewport-independent baseline presentation profile is intended to provide the highest interoperability and quality on HMDs (including mobile-powered HMDs).

This profile fulfils the basic requirements to support 3D audio and omnidirectional and 3D video. Both monoscopic and stereoscopic video are supported. The profile requires neither viewport-dependent delivery nor viewport-dependent decoding.

The profile also minimizes the options for basic interoperability.

11.1.2 ISO base media file format constraints

An ISOBMFF file for which the content author considers that the VR experience is included in this file using the technologies for the OMAF viewport-independent baseline presentation profile may be offered using the ISOBMFF file brand `'ompp'`.

For a file with `compatible_brands` containing a brand equal to `'ompp'` in `FileTypeBox`, the following constraints apply:

– The file shall conform to the `'iso9'` brand.

– If containing video, the file shall contain at least one track conforming to the HEVC-based viewport-independent OMAF video profile as specified in subclause 10.1.2.

– If containing audio, the file shall contain at least one track conforming to the OMAF 3D audio baseline profile as specified in subclause 10.2.2.

11.2 OMAF viewport-dependent baseline presentation profile

11.2.1 General

The OMAF viewport-dependent baseline presentation profile is intended to provide interoperability and quality on the HMDs that go beyond the viewport resolution achievable by the OMAF viewport-independent baseline presentation profile.

This profile fulfils requirements to support 3D audio and omnidirectional and 3D video. Both monoscopic and stereoscopic video are supported. The profile requires viewport-dependent delivery and rendering.

11.2.2 ISO base media file format constraints

An ISOBMFF file containing a VR experience using the technologies for the OMAF viewport-dependent baseline presentation profile may be offered using the ISOBMFF file brand `'ovdp'`.

For a file with `compatible_brands` containing a brand equal to `'ovdp'` in `FileTypeBox`, the following constraints apply:

– The file shall conform to the `'iso9'` brand.

– If containing video, the file shall contain at least one track conforming to the HEVC-based viewport-dependent OMAF video profile as specified in subclause 10.1.3.

– If containing audio, the file shall contain at least one track conforming to the OMAF 3D audio baseline profile as specified in subclause 10.2.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

 © ISO/IEC 2019 – All rights reserved

Annex A
(normative)
OMAF DASH schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:mpeg:mpegI:omaf:2017"
   xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
   elementFormDefault="qualified">
   <xs:element name="sphRegionQuality" type="omaf:SphRegionQualityType"/>
   <xs:complexType name="SphRegionQualityType">
      <xs:sequence>
         <xs:element name="qualityInfo" type="omaf:QualityInfoType" minOccurs="1" maxOccurs="255"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="shape_type" type="xs:unsignedByte" use="optional" default="0"/>
      <xs:attribute name="remaining_area_flag" type="xs:boolean" use="optional" default="0"/>
      <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional" default="0"/>
      <xs:attribute name="quality_ranking_local_flag" type="xs:boolean" use="optional" default="0"/>
      <xs:attribute name="quality_type" type="omaf:QualityType" use="required"/>
      <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>

   <xs:complexType name="QualityInfoType">
      <xs:attribute name="quality_ranking" type="xs:unsignedByte" use="required"/>
      <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
      <xs:attribute name="orig_width" type="xs:unsignedShort" use="optional"/>
      <xs:attribute name="orig_height" type="xs:unsignedShort" use="optional"/>
      <xs:attribute name="centre_azimuth" type="omaf:Range1" use="optional"/>
      <xs:attribute name="centre_elevation" type="omaf:Range2" use="optional"/>
      <xs:attribute name="centre_tilt" type="omaf:Range1" use="optional"/>
      <xs:attribute name="azimuth_range" type="omaf:HRange" use="optional"/>
      <xs:attribute name="elevation_range" type="omaf:VRange" use="optional"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>

   <xs:element name="twoDRegionQuality" type="omaf:twoDRegionQualityType"/>
   <xs:complexType name="twoDRegionQualityType">
      <xs:sequence>
         <xs:element name="twoDqualityInfo" type="omaf:twoDQualityInfoType" minOccurs="1"
maxOccurs="255"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="remaining_area_flag" type="xs:boolean" use="optional" default="0"/>
      <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional" default="0"/>
      <xs:attribute name="quality_ranking_local_flag" type="xs:boolean" use="optional"   default="0"/>
      <xs:attribute name="quality_type" type="omaf:QualityType" use="required"/>
      <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
   </xs:complexType>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
<xs:complexType name="twoDQualityInfoType">
  <xs:attribute name="quality_ranking" type="xs:unsignedByte" use="required"/>
  <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
  <xs:attribute name="orig_width" type="xs:unsignedShort" use="optional"/>
  <xs:attribute name="orig_height" type="xs:unsignedShort" use="optional"/>
  <xs:attribute name="left_offset" type="xs:unsignedShort" use="optional"/>
  <xs:attribute name="top_offset" type="xs:unsignedShort" use="optional"/>
  <xs:attribute name="region_width" type="xs:unsignedShort" use="optional"/>
  <xs:attribute name="region_height" type="xs:unsignedShort" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:element name="cc" type="omaf:CCType"/>
<xs:complexType name="CCType">
  <xs:sequence>
    <xs:element name="coverageInfo" type="omaf:coverageInfoType" minOccurs="1" maxOccurs="255"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="shape_type" type="xs:unsignedByte" use="optional" default="0"/>
  <xs:attribute name="view_idc_presence_flag" type="xs:boolean" use="optional" default="0"/>
  <xs:attribute name="default_view_idc" type="omaf:ViewType" use="optional"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="coverageInfoType">
  <xs:attribute name="view_idc" type="omaf:ViewType" use="optional"/>
  <xs:attribute name="centre_azimuth" type="omaf:Range1" use="optional" default="0"/>
  <xs:attribute name="centre_elevation" type="omaf:Range2" use="optional" default="0"/>
  <xs:attribute name="centre_tilt" type="omaf:Range1" use="optional" default="0"/>
  <xs:attribute name="azimuth_range" type="omaf:HRange" use="optional" default="23592960"/>
  <xs:attribute name="elevation_range" type="omaf:VRange" use="optional" default="11796480"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:attribute name="projection_type" type="omaf:listOfUnsignedByte"/>
<xs:simpleType name="listOfUnsignedByte">
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:unsignedByte"/>
    </xs:simpleType>
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>

<xs:attribute name="packing_type" type="omaf:OptionallistOfUnsignedByte"/>
<xs:simpleType name="OptionallistOfUnsignedByte">
  <xs:restriction>
    <xs:simpleType>
      <xs:list itemType="xs:unsignedByte"/>
    </xs:simpleType>
    <xs:minLength value="0"/>
  </xs:restriction>
</xs:simpleType>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

```
  <xs:attribute name="view_dimension_idc" type="omaf:viewDIdcType"/>
  <xs:simpleType name="viewDIdcType">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="7"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="Range1">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="-11796480"/>
      <xs:maxInclusive value="11796479"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Range2">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="-5898240"/>
      <xs:maxInclusive value="5898240"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="HRange">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="23592960"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="VRange">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="11796480"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="QualityType">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="15"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ViewType">
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="3"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved

Annex B
(normative)
DASH integration of media profiles

B.1 Video profiles

B.1.1 HEVC-based viewport-independent OMAF video profile

An instantiation of the HEVC-based viewport-independent OMAF video profile in DASH should be represented as one Adaptation Set, possibly with multiple Representations. If so, the Adaptation Set should provide the following signalling:

- `@codecs='resv.podv+erpv.hvc1.1.6.L93.B0'`
- `@mimeType='video/mp4 profiles="hevi"'`
- A supplemental descriptor or essential descriptor providing the frame packing arrangement may be used.

NOTE By the use of the restricted video scheme and the @profiles referring to this media profile, the DASH client has all information to identify if this media profile could be played back. For additional information, the supplemental descriptor is used to provide some details on the configuration of the contained Representations.

The concatenation of all DASH Segments of one Representation for HEVC viewport-independent baseline media profile shall conform to all the constraints specified in subclause 10.1.2.4.

Conformance to CMAF may be provided in addition by conforming to a HEVC CMAF video track as defined in ISO/IEC 23000-19:2018, Annex B.1.

In addition, for an Adaptation Set the following applies:

- The same frame packing format shall be used on all Representations in one Adaptation Set.
- The same coverage information shall be used on all Representations in one Adaptation Set.
- The same spatial resolution shall be used on all Representations in one Adaptation Set.

When the playback is intended to be started using another viewing orientation than that indicated by (`centre_azimuth`, `centre_elevation`, `centre_tilt`) equal to (0, 0, 0) relative to the global coordinate axes, a Representation containing initial viewing orientation metadata, as specified in subclause 7.7.4, shall be present and associated with all related media Representations as specified in subclause 8.2.3.

B.1.2 HEVC-based viewport-dependent OMAF video profile and AVC-based viewport-dependent OMAF video profile

B.1.2.1 General

This subclause applies to both the HEVC-based viewport-dependent OMAF video profile and the AVC-based viewport-dependent OMAF video profile.

When switching or accessing Representations at each Segment or Subsegment is relevant, the following DASH profiles include sufficient constraints:

- ISO base media file format live profile: `urn:mpeg:dash:profile:isoff-live:2011`
- ISO base media file format main profile: `urn:mpeg:dash:profile:isoff-main:2011`

When low latency considerations are relevant, the following DASH profiles provide tools to support efficient low latency services:

- ISO base media file format on demand profile: `urn:mpeg:dash:profile:isoff-on-demand:2011`
- ISO base media file format broadcast TV profile: `urn:mpeg:dash:profile:isoff-broadcast:2015`

It is recommended that DASH clients consuming low latency services support either or both of the above profiles in order to support the latency requirements.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 23090-2:2019

© ISO/IEC 2019 – All rights reserved