© ISO/IEC 2022 – All rights reserved

ISO/IEC 23008-12:2022/AMD 1:2022(E)

ISO/IEC JTC1/SC 29

Secretariat: JISC

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 1: Support for progressive rendering signalling and other improvements

CD stage

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

ISO copyright office

Case postale 56 • CH-1211 Geneva 20

Tel. + 41 22 749 01 11

Fax + 41 22 749 09 47

E-mail copyright@iso.org

Web www.iso.org

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

 © ISO/IEC 2022 – All rights reserved

Contents

Foreword ..... **iv**
6.5.37 Progressive derived image item information **2**
6.5.38 Single stream **2**
6.5.39 Camera extrinsic matrix **3**
6.5.40 Camera intrinsic matrix **5**
6.8.10 Progressive rendering entity group **6**
7.4.2 Sample-to-region-id-mapping sample group **7**
7.5.4 Region track and region annotations for an image sequence or video track **9**

© ISO/IEC 2022 – 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.

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 documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

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

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

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

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

A list of all parts in the ISO/IEC 23008 series can be found on the ISO and IEC websites.

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

 © ISO/IEC 2022 – All rights reserved

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 1: Support for progressive rendering signalling and other improvements

6.5

Replace the following in subclause 6.5.1 ("*General*")

Descriptive properties are non-essential, unless stated otherwise in their specification.

with

Descriptive properties should be marked as non-essential, unless stated otherwise in their specification or derived specification.

Replace the content of subclause 6.5.5.1 ("*Definition*" for "*Colour information"*) with the following

Box type:	`'colr'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	No
Quantity (per item):	Either at most one, or two with restriction described below

The definition of Colour Information provided in ISO/IEC 14496-12 applies.

In addition, following definitions specific to the use of colour information in image items also applies:

— When two `ColourInformationBoxes` are associated with an image item, one shall have a `colour_type` value of `'rICC'` or `'prof'` (providing either restricted or unrestricted ICC profiles respectively) and the other one shall have a `colour_type` value of `'nclx'` with `colour_primaries` equal to 2 and `transfer_characteristics` equal to 2 (2 indicating "unspecified", since these data are supplied by the ICC profile instead).

— When generating an image item from the content of a visual track, the order of `ColourInformationBoxes` in the `VisualSampleEntry` should be preserved in the `ItemPropertyAssociationBox`(es). Similarly, when creating a visual track from an image item, the order of boxes should be preserved.

— While in a visual track, the order of boxes may be important per ISO/IEC 14496-12, in this specification, the order is not relevant. Colour information with different values of `colour_type` are intended for different purposes. Colour information with a value of `colour_type` set to `'nclx'` is intended to be used for some processing such as colour conversion or image derivation, while colour information carrying an ICC profile is intended to be used for processes such as display matching.

© ISO/IEC 2022 – All rights reserved

Add the following new subclauses after subclause 6.5.36:

6.5.37 Progressive derived image item information

6.5.37.1 Definition

Box type:	`'prdi'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	No
Quantity (per item):	Zero or one for a derived image item

The progressive derived image item information property describes progressive rendering steps associated with a derived image item.

Each progressive rendering step specifies which input image items to use for the reconstruction of the derived image item and is described as a difference from the previous step.

NOTE the `ProgressiveDerivedImageItemInformationProperty` is intended to be used with derived image items using several input images.

6.5.37.2 Syntax

```
aligned(8) class ProgressiveDerivedImageItemInformationProperty
extends ItemFullProperty('prdi', version = 0, flags = 0){
   unsigned int(16) step_count;
   for (i=0; i < step_count; i++)
      unsigned int(16) item_count;
}
```

6.5.37.3 Semantics

`step_count` is the number of progressive steps for the associated derived image item.
`item_count` is the number of input image items added by the progressive step.

6.5.38 Single stream

6.5.38.1 Definition

Box type:	`'sstr'`
Property type:	Descriptive item property
Container:	`ItemPropertyContainerBox`
Mandatory (per item):	No
Quantity (per item):	Zero or one for a derived image item

The `SingleStreamProperty` indicates that the input image items to a derived image item, when concatenated, form a single bitstream that is conformant to the coding format of the input image items and is decodable with a single decoder. The order in which input image items are listed in the `'dimg'` item reference of a derived image item is the order in which the input image items are to be concatenated to obtain a conformant single bitstream. When an input image is a predictively coded image item, each one of its reference image items is included in the bitstream unless it has already been included in the bitstream.

NOTE The `SingleStreamProperty` is intended to be used with derived image items using several input images, for example, in progressive rendering.

 © ISO/IEC 2022 – All rights reserved

6.5.38.2 Syntax

```
aligned(8) class SingleStreamProperty
extends ItemFullProperty('sstr', version = 0, flags = 0){
}
```

6.5.39 Camera extrinsic matrix

6.5.39.1 Definition

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

The `CameraExtrinsicsMatrix` descriptive item property allows writers to communicate the spatial setup of camera(s). It is specified in the form of x, y, and z coordinates as well as an orientation in quaternion representation.

The coordinate system of the fields refers to the scene in which all the camera systems are positioned and can be identified, when coordinate systems occur in one file. The origin may be chosen to be at any point since the main intent of the camera extrinsics is to describe the relative positioning of the camera systems. It is recommended that the origin is placed either at one of the cameras in a multi-camera setup, or in the geometric centre.

The `flags` field is used to allow only specifying a sub-set of the extrinsic properties for common use-cases. The following flags are defined:

0x000001 this indicates that the position is signalled along the x axis.
0x000002 indicates that the position is signalled along the y axis.
0x000004 indicates that the position is signalled along the z axis.
0x000008 indicates that orientation is signalled.
0x000010 indicates that orientation elements are 32-bit integers (otherwise 16-bit integers).
0x000020 indicates that the coordinate system id is signalled.

Examples of use-cases are:

- Simple stereo pair (typically only requires a baseline):
  - use `flags` value 0x000001.
- Multiple cameras facing same direction (typically requires position in 2 or 3 axes):
  - use `flags` values from 0x000001 to 0x000007 (depending on which axes are non-zero).
- Panorama image collections (typically only requires orientation):
  - use `flags` values 0x000008 or 0x000018.
- 3D mapping image collections (typically requires both 3D position and orientation):
  - use `flags` values 0x00000F or 0x00001F.

`CameraExtrinsicsMatrix` instances with the same `id` value indicate that the associated image items were captured in the same coordinate system.

© ISO/IEC 2022 – All rights reserved

The variable *orientationPrecision* is set equal to (`flags` & 0x000010).

The values of the quaternion representation are computed as follows:

qX = `quat_x` / $2^{14 + orientationPrecision}$

qY = `quat_y` / $2^{14 + orientationPrecision}$

qZ = `quat_z` / $2^{14 + orientationPrecision}$

It is a requirement of bitstream conformance that:

$qX^2 + qY^2 + qZ^2 <= 1$

The fourth component of the quaternion representation, qW, is computed as follows:

$qW = \text{abs}(\text{sqrt}(1 - (qX^2 + qY^2 + qZ^2))$

NOTE: In the context of this specification qW is always positive. If a negative qW is desired, one can signal all three syntax elements, `quat_x`, `quat_y` and `quat_z`, with an opposite sign, which is equivalent.

6.5.39.2 Syntax

```
aligned(8) class CameraExtrinsicsMatrix
extends ItemFullProperty('cmex', version = 0, flags) {
   if (flags & 0x1) {
      signed int(32) pos_x;
   }
   if (flags & 0x2) {
      signed int(32) pos_y;
   }
   if (flags & 0x4) {
      signed int(32) pos_z;
   }
   if (flags & 0x8) {
      signed int(16 + (flags & 0x10)) quat_x;
      signed int(16 + (flags & 0x10)) quat_y;
      signed int(16 + (flags & 0x10)) quat_z;
   }
   if (flags & 0x20) {
      unsigned int(32) id;
   }
}
```

6.5.39.3 Semantics

`pos_x` specifies the x-coordinate of the location of the camera in µm. When not present, its value shall be inferred to be 0.

`pos_y` specifies the y-coordinate of the location of the camera in µm. When not present, its value shall be inferred to be 0.

`pos_z` specifies the z-coordinate of the location of the camera in µm. When not present, its value shall be inferred to be 0.

`quat_x` specifies the x component, qX, for the rotation of the camera using the quaternion representation. The range of `quat_x` shall be in the range of -2^N to 2^N, inclusive (where N is 14 + *orientationPrecision*). When not present, its value shall be inferred to be 0.

 © ISO/IEC 2022 – All rights reserved

`quat_y` specifies the y component, *qY*, for the rotation of the camera using the quaternion representation. The range of `quat_y` shall be in the range of -2^N to 2^N, inclusive (where N is 14 + *orientationPrecision*). When not present, its value shall be inferred to be 0.

`quat_z` specifies the z component, *qZ*, for the rotation of the camera using the quaternion representation. The range of `quat_z` shall be in the range of -2^N to 2^N, inclusive (where N is 14 + *orientationPrecision*). When not present, its value shall be inferred to be 0.

`id` specifies the coordinate system identifier. When not present, its value shall be inferred to be 0.

6.5.40 Camera intrinsic matrix

6.5.40.1 Definition

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

The `CameraIntrinsicsMatrix` descriptive item property allows writers to communicate the characteristics of the camera that captured the associated image item.

One general form of specifying the intrinsics matrix for a pinhole camera is as follows:

f_x	s	c_x
0	f_y	c_y
0	0	1

where:

f_x: horizontal focal length
f_y: vertical focal length
s: skew factor
c_x: principal point x
c_y: principal point y

NOTE 1 For most cameras, pixels are square and there is no skew. This corresponds to s being zero and f_x being equal to f_y.

The `flags` field is used to define the values of denominator and skew denominator.

The variable *denominator* is set equal to (1 << *denominatorShiftOperand*) where *denominatorShiftOperand* is equal to ((`flags` & 0x001F00) >> 8).

The variable *skewDenominator* is set equal to (1 << *skewDenominatorShiftOperand)* where *skewDenominatorShiftOperand* is equal to ((`flags` & 0x1F0000) >> 16).

The values of the above intrinsics matrix can be calculated as follows:

f_x = `focal_length_x` × `image_width` / *denominator*

f_y = `focal_length_y` × `image_height` / *denominator*

© ISO/IEC 2022 – All rights reserved

c_x = `principal_point_x` × `image_width` / *denominator*

c_y = `principal_point_y` × `image_height` / *denominator*

s = `skew_factor` / *skewDenominator*

where

`image_width` and `image_height` come from the `ImageSpatialExtentsProperty` associated with the image item.

NOTE 2 By specifying the focal lengths and principal point as normalized by image dimensions, this allows for the intrinsics matrix to be scale invariant. For a camera system without skew, this means that the same intrinsics matrix can be used even if the sensor uses pixel binning to output images with varying number of pixels.

6.5.40.2 Syntax

```
aligned(8) class CameraIntrinsicsMatrix
extends ItemFullProperty('cmin', version = 0, flags) {
   signed int(32) focal_length_x;
   signed int(32) principal_point_x;
   signed int(32) principal_point_y;
   if (flags & 1) {
      signed int(32) focal_length_y;
      signed int(32) skew_factor;
   }
}
```

6.5.40.3 Semantics

(`flags` & 1) equal to 0 indicates that simplified intrinsics (no skew, square pixels) are used. (`flags` & 1) equal to 1 indicates that full intrinsics are used.

`focal_length_x` specifies the horizontal focal length of the camera in image widths.

`focal_length_y` specifies the vertical focal length of the camera in image heights. When not present, the value shall be implied to be `focal_length_x` * `image_width` / `image_height`.

`principal_point_x` specifies the principal point x-coordinate in image widths.

`principal_point_y` specifies the principal point y-coordinate in image heights.

`skew_factor` specifies the camera system skew factor. When not present its value shall be implied to be 0.

6.8

Add the following new subclause after subclause 6.8.9:

6.8.10 Progressive rendering entity group

The progressive rendering entity group (with a `grouping_type 'prgr'`) signals a set of image items that can be used for a progressive rendering of one of these image items.

The semantics of the `'prgr'` entity group are that the image items included in a `'prgr'` entity group are listed in increasing quality order from the lowest quality to the highest quality. All the image items inside a `'prgr'` entity group shall correspond to similar images albeit with different quality levels. In

 © ISO/IEC 2022 – All rights reserved

this way, a first image item occurring earlier in the list than a second image item can be used as a temporary replacement of the second image item for a progressive rendering of this second image item.

The data corresponding to the image items included in a `'prgr'` entity group shall be stored in the same order as the one used for the image items inside the `'prgr'` entity group, such that a renderer progressively obtaining a file can perform a progressive display as item data becomes available.

A `'prgr'` entity group shall only contain image items, not tracks.

Image items of the same `'prgr'` entity group shall be members of the same `'altr'` entity group.

NOTE This requirement guarantees that legacy players without capability of processing `'prgr'` entity groups treat the image items as alternatives to be displayed.

7.4

Add the following new subclause after subclause 7.4.1:

7.4.2 Sample-to-region-id-mapping sample group

7.4.2.1 Definition

The sample-to-region-id-mapping (`'regm'`) sample grouping provides a flexible way to associate regions inside samples of a region track with region annotations. It provides the mapping between region identifiers in samples of the region track and region annotations carried in various types of containers: tracks, items or sample group descriptions.

The sample-to-region-id-mapping sample grouping should only be present in a region track, and may be ignored otherwise.

The number of entries (`entry_count`) declared in a `SampleToRegionIdMappingEntry` does not necessarily match 1:1 with the number of regions defined in associated samples of the region track. When there is no `regionID` value for an identifier of a region declared in a sample, this means that no annotation is associated with this region for the duration of this sample. Conversely, when there is no region declared in a sample with same value of identifier as a `regionID` value, this means that the corresponding annotations are not associated with any regions for the duration of this sample.

© ISO/IEC 2022 – All rights reserved

7.4.2.2 Syntax

```
class SampleToRegionIdMappingEntry()
extends SampleGroupDescriptionEntry('regm') {
   unsigned int(32) entry_count;
   for (i=1; i<= entry_count; i++) {
      unsigned int(32) regionID;
      unsigned int(16) association_count;
      for (i=0; i<association_count; i++) {
         unsigned int(5) reserved;
         unsigned int(3) annotation_container_type;
         unsigned int(32) annotation_reference_type;
         unsigned int(32) annotation_idx;
      }
   }
}
```

7.4.2.3 Semantics

`entry_count` is an integer that gives the number of entries in the mapping table, i.e., the number of regions in the mapping table.

`regionID` is an integer that specifies the identifier of the region.

`annotation_count` is an integer that specifies the number of annotations associated with the region.

`annotation_container_type` specifies the type of container carrying the annotation associated with the region. Following values for `annotation_container_type` are defined:

- 0: the annotation is carried in samples of the track or in the item identified by `annotation_reference_type` and `annotation_idx` in the track reference box. When the annotation is carried in samples of the track identified by `annotation_reference_type` and `annotation_idx`, the sample in that track temporally aligned, or nearest preceding in the media presentation timeline, comprises the annotation applying to the corresponding sample of the track referenced by the region track with a track reference `'cdsc'`.
- 1: the annotation is the `SampleGroupDescriptionEntry` corresponding to the index `annotation_idx` in the `SampleGroupDescriptionBox` with `grouping_type` equal to `annotation_reference_type`.
- 2-7: reserved.

`annotation_reference_type` specifies the four-character code type of the container carrying the annotation depending on the `annotation_container_type` value. If the value of `annotation_container_type` equals 0, it specifies the `reference_type` of the `TrackReferenceTypeBox` in the `TrackReferenceBox` containing the identifier of the entity (i.e., track or item) containing the annotation associated with the region. If the value of `annotation_container_type` equals 1, it specifies the `grouping_type` of the `SampleGroupDescriptionBox` containing the annotation associated with the region. The meaning of `annotation_reference_type` is undefined for other values of `annotation_container_type`.

`annotation_idx` specifies the index to retrieve the container carrying the annotation depending on the `annotation_container_type` value. If the value of `annotation_container_type` equals 0, it specifies the index of the track reference of type `annotation_reference_type` identifying the track or item from which to retrieve the annotation associated with the region. If the value of `annotation_container_type` equals 1, it specifies the index of the `SampleGroupDescriptionEntry` in the `SampleGroupDescriptionBox` with `grouping_type` equal to `annotation_reference_type` from which to retrieve the annotation associated with the

 © ISO/IEC 2022 – All rights reserved

region. The meaning of `annotation_idx` is undefined for other values of `annotation_container_type`.

7.5

Add the following new subclause after subclause 7.5.3:

7.5.4 Region track and region annotations for an image sequence or video track

7.5.4.1 Definition

A metadata track with a sample entry `'rgan'` is a region track whose samples define one or more regions inside images carried in samples of an associated image sequence or video track (also denoted source track in the following).

The region track is associated with the source track inside which the regions are defined using a track reference of type `'cdsc'` from the region track to the source track.

A region track allows associating a region annotation with each region it defines inside a sample of the associated source track. A region annotation may be an item (e.g., an image item, a metadata item), samples from another track, or a `SampleGroupDescriptionEntry`.

Each region in a sample of a region track is associated with a region identifier. The scope of region identifiers is the sample. The region identifier can be used to associate an annotation with a region in a sample via an association map defined in a sample-to-region-id-mapping sample group. In such case, region identifier values in samples and in the sample group have the same value space.

Different regions in a same sample or in consecutive samples may have the same region identifier. When multiple regions in the same sample have the same region identifier, this indicates that each of these regions is associated with the same set of region annotations.

A `SampleGroupDescriptionEntry` should only be associated with a region defined by a region track when the property value for the region differs from the matching property value, explicit or implicit, for the whole sample of the associated source track.

The geometries of the regions described in the region track are specified in the data of the samples of the region track. Samples of the region track should be time-aligned with corresponding samples of the source track in the media presentation timeline.

These geometries define the shape, position and size of the regions inside a reference space that is mapped to the samples of the source track with which the region track is associated after any transformation defined in the source track is applied to the samples of the source track.

The geometry of a region inside the sample of the associated source track is obtained after applying the implicit resampling caused by the difference between the size of the reference space and the size of the sample of the associated source track.

The geometry of a region described by the samples of the region track can be represented either by:

— a point,

— a polyline,

© ISO/IEC 2022 – All rights reserved

— a rectangle,

— an ellipse,

— a polygon,

— a mask stored in the sample of another track or in an image item, or,

— a mask defined in the data of the sample.

Line segments for both polygons and polylines shall not cross (including the implicit closing line for polygons).

The pixels of the sample of the associated source track that are part of a region depend on the geometry of the region as follows:

— When the geometry of a region is represented by a point, the pixel located at this point, if it exists, is part of the region.

— When the geometry of a region is represented by a rectangle, an ellipse, or a polygon, the pixels that are inside (including the boundary) of the rectangle, ellipse, or polygon are part of the region.

— When the geometry of a region is represented as a mask defined in the data of the sample of the region track (i.e., when `geometry_type` equals 5), the pixels of the sample of the associated source track corresponding to pixels with the value 1 in the mask image are part of the region.

— When the geometry of a region is represented by a polyline, the pixels are part of the region when they would be coloured or partly coloured (e.g. when using anti-aliasing) if a one-pixel-wide line were drawn along the polyline.

— When the geometry of a region is represented by a mask stored in the sample of another track or in an image item (i.e., when `geometry_type` equals 4), the image item or the sample containing the mask is the timed-aligned sample of the track identified by a track reference of type `'mask'` from the region track to the track or image item containing the mask. The track or image item containing the mask shall be one of the following:

    — An image item or a track that is encoded in monochrome format (i.e. 4:0:0 chroma format).

    — An image item or a track that is encoded in colour. In such a case, it shall be encoded in a colour format with a luma plane and chroma planes (e.g. as 4:2:0 YCbCr). Since only the luma plane is relevant, the chroma planes should be ignored.

A region may be empty if it falls entirely outside the image carried in a sample. An empty region should be ignored.

The mask contained in a sample of another track or in an image item applies to a region after performing the implicit resampling caused by the difference between the size of the region documented by width and height and the size of the sample or image item containing the mask.

When the mask is stored as an image item or as a sample of a track, transformations associated with the image item or track may apply to the mask before it is used to define the geometry of a region inside the referenced image item.

 © ISO/IEC 2022 – All rights reserved

— The following semantics apply to a mask stored as a mask item, image item or sample of another track:the maximum sample value (e.g. 255 for 8-bit sample values) in the mask image means that the corresponding pixel in the sample of the associated track is part of the region;

— the minimum sample value (i.e. 0) in the mask image means that the corresponding pixel in the sample of the associated track is not part of the region;

— other sample values (e.g. 1 to 254 for 8-bit sample values) is representative of a probability that the corresponding pixel in the sample of the associated track is part of the region. Higher pixel values represent higher probability values.

NOTE 1 The term "sample value" used above is to be interpreted as "luma sample value" if the mask is stored as an image item or sample encoded with separate luma and chroma planes.

NOTE 2 Non-zero mask pixel values can have application-dependent semantics. For example, they can represent confidence scores for each pixel of an AI-generated mask, or saliency score indication per pixel in the defined region.

An alpha plane auxiliary track or image item may be referenced as a mask by a region track. In such case, the reference space size should be equal to the alpha plane size and the coordinates (x, y) of the region should be (0, 0).

7.5.4.2 Sample entry

7.5.4.2.1 Definition

Sample entry type:	`'rgan'`
Container:	`SampleDescriptionBox`
Mandatory:	Yes
Quantity:	One or more sample entries may be present

The sample entry documents the reference space inside which regions are defined.

The reference space is defined as a 2D coordinate system with the origin (0,0) located at the top-left corner and a maximum size defined by `reference_width` and `reference_height`; the x-axis is oriented from left to right and the y-axis from top to bottom.

7.5.4.2.2 Syntax

```
aligned(8) class RegionTrackConfigBox () extends FullBox ('rgaC', version=0,
flags=0) {
   unsigned int (7) reserved = 0;
   unsigned int (1) field_length_size;
   unsigned int((field_length_size + 1) * 16) reference_width;
   unsigned int((field_length_size + 1) * 16) reference_height;
}

aligned(8) class RegionSampleEntry
extends MetadataSampleEntry ('rgan'){
   RegionTrackConfigBox config; // mandatory
}
```

7.5.4.2.3 Semantics

`field_length_size` is a boolean indicating the length of various fields. The value equal to 0 specifies that the length of a field is 16 bits. The value equal to 1 specifies that the length of a field is 32 bits.

© ISO/IEC 2022 – All rights reserved

`reference_width`, `reference_height` specify, in pixel units, the width and height, respectively, of the reference space inside which the regions are placed.

7.5.4.3 Sample format

7.5.4.3.1 Definition

This subclause defines the sample format for region track. A sample of a region track defines one or more regions.

© ISO/IEC 2022 – All rights reserved

7.5.4.3.2 Syntax

```
aligned (8) class RegionSample {
   unsigned int field_size = ((RegionTrackConfigBox.field_length_size & 1) + 1) *
16;
// this is a temporary, non-parsable variable
   unsigned int(32) region_count;
   for (r=0; r < region_count; r++) {
      unsigned int(32) region_identifier;
      unsigned int(8) geometry_type;
      if (geometry_type == 0) {
         // point
         signed int(field_size) x;
         signed int(field_size) y;
      }
      else if (geometry_type == 1) {
         // rectangle
         signed int(field_size) x;
         signed int(field_size) y;
         unsigned int(field_size) width;
         unsigned int(field_size) height;
      }
      else if (geometry_type == 2) {
         // ellipse
         signed int(field_size) x;
         signed int(field_size) y;
         unsigned int(field_size) radius_x;
         unsigned int(field_size) radius_y;
      }
      else if (geometry_type == 3 || geometry_type == 6) {
         // polygon or polyline
         unsigned int(field size) point_count;
         for (i=0; i < point_count; i++) {
            signed int(field_size) px;
            signed int(field_size) py;
         }
      }
      else if (geometry_type == 4) {
         // referenced mask
         signed int(field_size) x;
         signed int(field_size) y;
         unsigned int(field_size) width;
         unsigned int(field_size) height;
         unsigned int(field_size) track_mask_idx;
      }
      else if (geometry_type == 5) {
         // inline mask
         signed int(field_size) x;
         signed int(field_size) y;
         unsigned int(field_size) width;
         unsigned int(field_size) height;
         unsigned int(8) mask_coding_method;
         if (mask_coding_method != 0)
            unsigned int(32) mask_coding_parameters;
         bit(8) data[];
      }
   }
}
```

7.5.4.3.3 Semantics

`region_count` specifies the number of regions defined in the sample.

© ISO/IEC 2022 – All rights reserved

`region_identifier` specifies the identifier of the region.

`geometry_type` specifies the type of the geometry of a region. The following values for geometry_type are defined:

- 0: the region is described as a point.
- 1: the region is described as a rectangle.
- 2: the region is described as an ellipse.
- 3: the region is described as a polygon.
- 4: the region is described as a mask defined in a referenced image item or in a sample of a referenced track.
- 5: the region is described as a mask defined inside the data of this sample.
- 6: the region is described as a polyline.

Other values are reserved.

`x`, `y` specify the coordinates of the point composing the region relatively to the reference space when its geometry is a point. `x`, `y` specify the top, left corner of the region relatively to the reference space when its geometry is a rectangle or a mask. `x`, `y` specify the centre of the region relatively to the reference space when its geometry is an ellipse. The value (`x` = 0, `y` = 0) represents the position of the top-left pixel in the reference space.

NOTE 1 Negative values for the `x` or `y` fields enable to specify points, top-left corners, and/or centres that are outside the image. This can be useful for updating region annotations during edition.

`width`, `height` specify, relatively to the reference space, the width and the height of the region when its geometry is a rectangle or a mask. When `geometry_type` equals 4, the value 0 indicates that the corresponding `width` or `height` value is provided by the `ImageSpatialExtentsProperty` associated with the item containing the mask or the `width` or `height` of the track containing the mask. When `geometry_type` does not equal 4, the value 0 is reserved.

`radius_x` specifies, relatively to the reference space, the radius on the x-axis of the region when its geometry is an ellipse.

`radius_y` specifies, relatively to the reference space, the radius on the y-axis of the region when its geometry is an ellipse.

`point_count` is the number of points contained in a polygon or a polyline.

NOTE 2 A polygon specifying the geometry of a region is always closed and therefore there is no need to repeat the first point of the polygon as the ending point of the polygon.

`px`, `py` specify the coordinates of the points composing the polygon or the polyline relatively to the reference space. The value (`px` = 0, `py` = 0) represents the position of the top-left pixel in the reference space.

`track_mask_idx` specifies the index of the track reference of type `'mask'` referring to the track from which to retrieve the mask to apply. The sample in that track from which mask data is retrieved is the one that is temporally aligned with the current sample in the source track or the nearest preceding one in the media presentation timeline. The first track reference has the index value 1; the value 0 is reserved.

`mask_coding_method` indicates the coding method applied on the mask contained in `data`. The following values are defined:

- 0: No mask encoding scheme is applied.
- 1: Mask is compressed with deflate() as defined in IETF RFC 1951.

Other values are reserved.

`mask_coding_parameters` indicates additional encoding parameters needed for successfully processing the coded mask data. When `mask_coding_method` is equal to 1, `mask_coding_parameters` indicates the number of bytes in the coded mask array `data`. The

 © ISO/IEC 2022 – All rights reserved

value of `mask_coding_parameters` is reserved when the value of `mask_coding_method` is greater than 1.

`data` contains the coded or uncompressed representation of a mask that contains the pixels for an inline mask in raster-scan order. Each pixel is represented using a single bit and 8 pixels are packed in one byte. Byte packing shall be in big-endian order. No padding shall be put at the end of each line if the `width` of the mask is not a multiple of 8 pixels. Only the last data byte shall be padded with bits set to 0.

7.5.4.4 Region annotation for a track

A region annotation may consist in metadata or images or audio samples carried in image items, samples of metadata or video or audio track, or properties from a sample group associated with one or more regions of an image carried in a sample of an image sequence or video track.

A region annotation may be associated with one or more regions of a sample of an image sequence or video track by:

— describing in a sample of a region track the geometry of these one or more regions;

— associating the region track with the image sequence or video track it describes using the `'cdsc'` (content describes) track reference from the region track to the image sequence or video track;

— associating tracks or items carrying the region annotation with the region track using a track reference of type `'anot'` from the region track to the image items or the tracks, and/or defining the `SampleGroupDescriptionBox`es containing the region annotation; and

— defining a sample-to-region-id-mapping sample grouping (`'regm'`) in the region track providing the mapping between the region identifiers and the region annotations.

The region annotation applies to each region described in time-aligned sample of the region track individually.

Annex H.2

Replace the following in subclause H.2.1 ("*Definition*")

The concatenation of the contents of the optional JPEG configuration box (the `JPEGprefix` bytes) with the extents of the JPEG image item shall conform to the specification for a JPEG compressed image as defined in ISO/IEC 10918-1, starting with the SOI (start of image) marker and ending with the EOI (end of image) marker.

With

The concatenation of the contents of the optional JPEG configuration item property (the `JPEGprefix` bytes) with the extents of the JPEG image item shall conform to the specification for a JPEG compressed image as defined in ISO/IEC 10918-1, starting with the SOI (start of image) marker and ending with the EOI (end of image) marker.

Replace the following in subclause H.2.2 ("*JPEG configuration item property*")

Each JPEG image item may have an associated configuration property.

© ISO/IEC 2022 – All rights reserved

with

Each JPEG image item may have an associated item property of type `'jpgC'` with the syntax identical to the `JPEGConfigurationBox` as defined in H.2.3.

NOTE The JPEG configuration item property extends `ItemProperty` which is equivalent to `Box`.

Rename the subclause H.2.3 ("*Syntax*") to "JPEG configuration box syntax".

Annex L.2.2.1

Replace entire NOTE 3 in subclause L.2.2.1.2 ("*Image item of type 'vvc1'*") with the following:

NOTE 3 ISO/IEC 23090-3 requires that a VVC decoder conforming to a profile at a specific level and specific tier is able to decode the first picture of a VVC bitstream when all of the following applies:

- at least one of the following:
  - the VVC decoder conforms to the Main 10 Still Picture profile and the bitstream conforms to the Main 10 profile;
  - the VVC decoder conforms to the Main 10 4:4:4 Still picture profile and the bitstream conforms to the Main 10 profile or the Main 10 4:4:4 profile;
  - the VVC decoder conforms to the Main 12 Still Picture profile and the bitstream conforms to one of the Main 10, Main 10 4:4:4, Main 12, or Main 12 Intra profiles;
  - the VVC decoder conforms to the Main 12 4:4:4 Still Picture profile and the bitstream conforms to one of the Main 10, Main 10 4:4:4, Main 12, Main 12 Intra, Main 12 4:4:4, or Main 12 4:4:4 Intra profiles;
  - the VVC decoder conforms to the Main 16 4:4:4 Still Picture profile and the bitstream conforms to one of the Main 10, Main 10 4:4:4, Main 12, Main 12 Intra, Main 12 4:4:4, Main 12 4:4:4 Intra, Main 16 4:4:4, or Main 16 4:4:4 Intra profiles;
- the VVC bitstream conforms to a tier that is lower than or equal to the specified tier;
- the VVC bitstream conforms to a level that is not level 15.5 and is lower than or equal to the specified level;
- the first picture of the VVC bitstream is an IRAP picture or a GDR picture with ph_recovery_poc_cnt equal to 0, is in an output layer, and has ph_pic_output_flag equal to 1.

 © ISO/IEC 2022 – All rights reserved