**ISO/IEC 23008-12:2024/AMD 1:2024(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 tone map derived image items and other improvements**

DAM 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.

Contents

[Foreword v](#_Toc183680031)

[1 Coding Constraints box related changes 1](#_Toc183680032)

[2 New tone-map derivation item 2](#_Toc183680033)

[6.6.2.4 Tone-map derivation 2](#_Toc183680034)

[6.6.2.4.1 Definition 2](#_Toc183680035)

[6.6.2.4.2 Syntax 3](#_Toc183680036)

[6.6.2.4.3 Semantics 4](#_Toc183680037)

[10.2.6 'tmap' brand 4](#_Toc183680038)

[J.6 Tone-map derivation 4](#_Toc183680039)

[3 New colour format enhancement derived image item 7](#_Toc183680040)

[6.6.2.5 Colour format enhancement derivation 7](#_Toc183680041)

[6.6.2.5.1 Definition 7](#_Toc183680042)

[6.6.2.5.2 Syntax 8](#_Toc183680043)

[6.6.2.5.3 Semantics 8](#_Toc183680044)

[4 New constrained extents grid property 9](#_Toc183680045)

[6.5.41 Constrained Extents Grid Property 9](#_Toc183680046)

[6.5.41.1 Definition 9](#_Toc183680047)

[6.5.41.2 Syntax 10](#_Toc183680048)

[6.5.41.3 Semantics 11](#_Toc183680049)

[5 Stereo related signaling 11](#_Toc183680050)

[5.1 Clarifications on camera extrinsic and intrinsic properties 11](#_Toc183680051)

[6.5.42 Disparity adjustment information 12](#_Toc183680052)

[6.5.42.1 Definition 12](#_Toc183680053)

[6.5.42.2 Syntax 13](#_Toc183680054)

[6.5.42.3 Semantics 13](#_Toc183680055)

[6.5.43 Stereo aggressors item property 13](#_Toc183680056)

[6.5.43.1 Definition 13](#_Toc183680057)

[6.5.43.2 Syntax 14](#_Toc183680058)

[6.5.43.3 Semantics 14](#_Toc183680059)

[6.8.10 'stem' entity grouping 15](#_Toc183680060)

[7 New HDR signaling aligning with ISO 22028-5 15](#_Toc183680061)

[8.1.44 Reference viewing environment 15](#_Toc183680062)

[8.1.44.1 Definition 15](#_Toc183680063)

[8.1.44.2 Syntax 16](#_Toc183680064)

[8.1.44.3 Semantics 16](#_Toc183680065)

[8.1.45 Nominal Diffuse White 16](#_Toc183680066)

[8.1.45.1 Definition 16](#_Toc183680067)

[8.1.45.2 Syntax 17](#_Toc183680068)

[8.1.45.3 Semantics 17](#_Toc183680069)

[9 Unified identifier handling clarifications 17](#_Toc183680070)

[10 Overview images 17](#_Toc183680071)

[6.4.10 Overview images 17](#_Toc183680072)

[6.8.11 Image Pyramid Entity Group 18](#_Toc183680073)

[6.8.11.1 Definition 18](#_Toc183680074)

[6.8.11.2 Syntax 21](#_Toc183680075)

[6.8.11.3 Semantics 21](#_Toc183680076)

[10.1 Tiled image item 21](#_Toc183680077)

[6.12 Tiled image items 21](#_Toc183680078)

[9 Region partition group 27](#_Toc183680079)

[6.12.5 Region Partition Group 27](#_Toc183680080)

[6.12.5.1 Definition 27](#_Toc183680081)

[6.12.5.2 Syntax 29](#_Toc183680082)

[6.12.5.3 Semantics 29](#_Toc183680083)

[10 Change all mention of SingleItemTypeReferenceBox to include SingleItemTypeReferenceBoxLarge 29](#_Toc183680084)

[11 Add T.35 metadata to Annex A 30](#_Toc183680085)

[A.6 ITU-T T.35 metadata 31](#_Toc183680086)

[12 Clarify that items may share auxiliary images, thumbnails and metadata 31](#_Toc183680087)

[13 Merge region item 31](#_Toc183680088)

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](http://www.iso.org/directives) or [www.iec.ch/members\_experts/refdocs](http://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](http://www.iso.org/patents)) or the IEC list of patent declarations received (see [patents.iec.ch](https://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](http://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](https://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](https://www.iso.org/members.html) and [www.iec.ch/national-committees](http://www.iec.ch/national-committees).

Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format — Amendment 1: Support for tone map derived image items and other improvements

# Coding Constraints box related changes

In clause 7.2.3.4, renumber NOTE2 to NOTE3.

In clause 7.2.3.4, add the following text after NOTE1:

NOTE 2: When a track contains inter-predicted images and the value of all\_ref\_pics\_intra is equal to 0, it is possible for inter-predicted images to be derived from non-intra coded images. In such cases, derived specifications can suggest guidelines for the frequency of sync samples.

Add the following text as a new subclause after subclause 7.2.3.4:

7.2.3.5 Recommendations for CodingConstraintsBox

Encoding image sequences complying with the constraint that either all samples are sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox can be set to one is suggested in the following cases:

1. For ensuring compatibility with players implementing codec brands in annexes B, E, L and M specified in an earlier edition of this document, which required all samples to be sync samples or to have the all\_ref\_pics\_intra field in the CodingConstraintsBox to be equal to 1.
2. In applications and usages where fast random access is essential, the constraint ensures random access to any image in the image sequence by decoding at most up to two images.
3. In image sequences that have an edit list causing backward playback, the constraint makes backward playback possible with an approach that each displayed image is obtained by decoding a compliant bitstream of up to two images.

In clause B.3.2, replace the following paragraph

For a track containing an HEVC image sequence, either all samples shall be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 7.2.3 shall be set to one.

with the following text

For a track containing an HEVC image sequence, either all samples should be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 7.2.3 should be set to one.

NOTE: Clause 7.2.3.5 contains recommendations that are important for backwards compatibility.

In clause E.3.2, replace the following paragraph

For a track containing an AVC image sequence, either all samples shall be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 1 shall be set to one.

with the following text

For a track containing an AVC image sequence, either all samples should be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 7.2.3 should be set to one.

NOTE: Clause 7.2.3.5 contains recommendations that are important for backwards compatibility.

In clause L.3.2, replace the following paragraph

For a track containing an VVC image sequence, either all samples shall be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in ‎1 shall be set to one.

with the following text

For a track containing an VVC image sequence, either all samples should be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 7.2.3 should be set to one.

NOTE: Clause 7.2.3.5 contains recommendations that are important for backwards compatibility.

In clause M.3.2, replace the following paragraph

For a track containing an EVC image sequence, either all samples shall be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in ‎1 shall be set to one.

with the following text

For a track containing an EVC image sequence, either all samples should be sync samples or the all\_ref\_pics\_intra field in the CodingConstraintsBox specified in 7.2.3 should be set to one.

NOTE: Clause 7.2.3.5 contains recommendations that are important for backwards compatibility.

# New tone-map derivation item

Add the following new subclause after subclause 6.6.2.3:

## Tone-map derivation

### 6.6.2.4.1 Definition

An item with an item\_type value of 'tmap' defines a derived image item whose reconstructed image is formed from one base input image and a secondary input image that will be referred to as a gain map input image.

The input images are given by the SingleItemTypeReferenceBox/ SingleItemTypeReferenceBoxLarge of type 'dimg' for this derived image item within the ItemReferenceBox. In the SingleItemTypeReferenceBox/ SingleItemTypeReferenceBoxLarge of type 'dimg', the value of from\_item\_ID identifies the derived image item of type 'tmap', the value of reference\_count shall be equal to 2, and the values of to\_item\_ID identify the input images, of which the first shall be the base input image and the second shall be the gain map input image. The gain map input image may have different dimensions than the base as documented in ISO 21496-1 section 4.2.

Reconstruction is done by applying the gain map to the base image according to ISO 21496-1 section 6. As described in ISO 21496-1 section 6.3, the gain map may be scaled by a weight during application to adjust for local viewing conditions.

The base input image shall be associated with a 'colr' item property. This corresponds to the colorimetry metadata described in ISO 21496-1 section 5.3.1.

The gain map input image shall be associated with a 'colr' item property of type 'nclx' which indicates any transformations that the encoder has done to improve compression. In this item property, colour\_primaries and transfer\_characteristics shall be set to 2. The full\_range\_flag may be set to either 1 or 0. If the full\_range\_flag is 0, the reader shall clip the gain map pixel values to the logical range 0.0 to 1.0 after applying matrix\_coefficients and full\_range\_flag.

A 'tmap' derived image item shall be associated with a 'colr' item property. This corresponds to the colorimetry metadata described in ISO 21496-1 section 5.3.2, which describes the colour properties of the reconstructed image if the gain map input item is fully applied according to ISO 21496-1 section 6.3.

When a 'tmap' derived image item is the input to another derived image item, that derived image item shall treat the reconstructed image of the 'tmap' derived image item as if the gain map has been fully applied and has the colour properties of the 'colr' item property associated with the 'tmap' derived image item.

The base input image and the 'tmap' derived image item should be associated with 'clli' item properties as appropriate to further document the optimal viewing conditions of each representation.

A 'tmap' derived image item should be associated with a PixelInformationProperty item property. This property provides a hint to decoders on the approximate amount of colour resolution available after fully applying the gain map.

The number of channels in the gain map input item may be different than the number of channels in the gain\_map\_metadata in the body of the 'tmap' derived image item. If the gain map input item is single channel while the gain\_map\_metadata signals multi-channel, the gain map input item shall be treated as if it has three identical colour channels. If the gain map input item is multi-channel while the gain\_map\_metadata signals single channel, the gain\_map\_metadata shall be treated as if it is multi-channel with identical values for all channels.

EDITORS NOTE: Sync up with 21496-1 project. Given that the metadata payload is defined in 21496-1 now, these constraints are already specified in 21496-1 if I remember correctly. It may be better to not re-specify them here and leave these edge cases to 21496-1.

The gain map input image should be marked as hidden by setting (flags & 1) equal to 1 in its ItemInfoEntry.

The requirements of clause 10.2.6 apply to any file containing a tone-map derived image item.

NOTE 1: Backwards compatibility with parsers that do not support the tone-map derivation can be achieved by placing the base input image item and the 'tmap' derived image item in an 'altr' entity group.

NOTE 2: The ToneMapImage has a version field. This is separate from the GainMapVersion field in the GainMapMetadata structure in ISO 21496-1 clause C.2. The ToneMapImage version field allows for versioning of how the tone-map derivation is done in this specification.

### Syntax

aligned(8) class ToneMapImage {

unsigned int(8) version = 0;

if (version == 0) {

bit(8) gain\_map\_metadata[];

}

}

### Semantics

version shall be equal to 0. Readers shall not process a ToneMapImage with an unrecognized version number.

gain\_map\_metadata specifies the gain map metadata as defined in ISO 21496-1 clause C.2 in the GainMapMetadata structure.

Add the following new subclause after subclause 10.2.5:

## 'tmap' brand

#### Definition

This brand enables file players to identify and decode HEIF files containing tone-map derived image items. When present, this brand shall be among the brands included in the compatible\_brands array of the FileTypeBox.

#### Requirements on files

A file containing the 'tmap' brand in the compatible\_brands array of the FileTypeBox shall contain one or more tone-map derived image items.

#### Requirements on readers

Readers shall support the following:

* Support for tone-map derived image items
* Within the entity groups, support for EntityToGroupBox with grouping\_type equal to 'altr'

Add the following new subclause after subclause J.6:

## Tone-map derivation

This example illustrates how a tone-map derived image item can be stored in a file in a backwards compatible manner using an 'altr' entity group. Both the base image item and the gain map image items are tiled using grid derived image items. The base image and gain map are explicitly marked as having zero rotation.

FileTypeBox 'ftyp': major-brand='heic', compatible-brands='tmap, mif1, heic'

MetaBox 'meta': (container)

HandlerBox 'hdlr': 'pict'

PrimaryItemBox 'pitm': item\_ID=1;

ItemInfoBox 'iinf': entry\_count=11  
 // base image  
 1) 'infe': item\_ID=1, item\_type='grid';  
 2) 'infe': item\_ID=2(Hidden), item\_type='hvc1’;  
 3) 'infe': item\_ID=3(Hidden), item\_type='hvc1';  
 4) 'infe': item\_ID=4(Hidden), item\_type='hvc1';  
 5) 'infe': item\_ID=5(Hidden), item\_type='hvc1';  
 // gain map  
 6) 'infe': item\_ID=6(Hidden), item\_type='grid';  
 7) 'infe': item\_ID=7(Hidden), item\_type='hvc1';  
 8) 'infe': item\_ID=8(Hidden), item\_type='hvc1';  
 9) 'infe': item\_ID=9(Hidden), item\_type='hvc1';  
 10)'infe': item\_ID=10(Hidden), item\_type='hvc1';  
 // tmap  
 11)'infe': item\_ID=11, item\_type='tmap';

ItemLocationBox 'iloc': item\_count=11  
 item\_ID=1, construction\_method=1, extent\_count=1, extent\_offset=P1, extent\_length=Q1;  
 item\_ID=2, extent\_count=1, extent\_offset=P2, extent\_length=Q2;  
 item\_ID=3, extent\_count=1, extent\_offset=P3, extent\_length=Q3;  
 item\_ID=4, extent\_count=1, extent\_offset=P4, extent\_length=Q4;  
 item\_ID=5, extent\_count=1, extent\_offset=P5, extent\_length=Q5;  
 item\_ID=6, construction\_method=1, extent\_count=1, extent\_offset=P6, extent\_length=Q6;  
 item\_ID=7, extent\_count=1, extent\_offset=P7, extent\_length=Q7;  
 item\_ID=8, extent\_count=1, extent\_offset=P8, extent\_length=Q8;  
 item\_ID=9, extent\_count=1, extent\_offset=P9, extent\_length=Q9;  
 item\_ID=10, extent\_count=1, extent\_offset=P10, extent\_length=Q10;  
 item\_ID=11, extent\_count=1, extent\_offset=P11, extent\_length=Q11;

ItemReferenceBox 'iref':

// base image grid of 4 tiles

referenceType='dimg', from\_item\_ID=1, reference\_count=4,

to\_item\_ID=2,

to\_item\_ID=3,

to\_item\_ID=4;

to\_item\_ID=5;

// A (hidden) gain map is composed of 4 tiles in the grid

referenceType='dimg', from\_item\_ID=6, reference\_count=4,

to\_item\_ID=7,

to\_item\_ID=8,

to\_item\_ID=9;

to\_item\_ID=10;

// tmap  
 referenceType='dimg', from\_item\_ID=11, ref\_count=2,

to\_item\_ID=1, // base

to\_item\_ID=6; // gain map

ItemPropertiesBox 'iprp':  
 ItemPropertyContainerBox 'ipco':

// Base image, item 1, grid of tiles  
 'ispe'  
 'pixi'

'colr'  
 'clli'

// tiles, items 2-5

'ispe'

'hvcC'

// Gain map, item 6, grid of tiles

'ispe'  
 'pixi'

'colr'

// tiles, items 7-10

'ispe'

'hvcC'

// tmap, item 11  
 'colr'

'clli'

'pixi'

// Explicitly mark orientation of base and gain map  
 'irot': angle=0

ItemPropertyAssociation 'ipma': entry\_count=11  
 1) item\_ID=1, association\_count=5 // base  
 essential=0, property\_index=1;  
 essential=0, property\_index=2;  
 essential=1, property\_index=3;

essential=0, property\_index=4;

essential=1, property\_index=15;

2) item\_ID=2, association\_count=4 // base - tile 1

essential=1, property\_index=6;  
 essential=0, property\_index=5;  
 essential=0, property\_index=2;  
 essential=1, property\_index=3;

3) item\_ID=3, association\_count=4 // base - tile 2

essential=1, property\_index=6;  
 essential=0, property\_index=5;  
 essential=0, property\_index=2;  
 essential=1, property\_index=3;

4) item\_ID=4, association\_count=4 // base - tile 3

essential=1, property\_index=6;  
 essential=0, property\_index=5;  
 essential=0, property\_index=2;  
 essential=1, property\_index=3;

5) item\_ID=5, association\_count=4 // base - tile 4

essential=1, property\_index=6;  
 essential=0, property\_index=5;  
 essential=0, property\_index=2;  
 essential=1, property\_index=3;

6) item\_ID=6, association\_count=4 // gain map  
 essential=0, property\_index=7;  
 essential=0, property\_index=8;  
 essential=1, property\_index=9;  
 essential=1, property\_index=15;

7) item\_ID=7, association\_count=4 // gain map - tile\_1

essential=1, property\_index=11;

essential=0, property\_index=10;  
 essential=0, property\_index=8;

essential=1, property\_index=9;

8) item\_ID=8, association\_count=4 // gain map - tile\_2

essential=1, property\_index=11;

essential=0, property\_index=10;  
 essential=0, property\_index=8;

essential=1, property\_index=9;

9) item\_ID=9, association\_count=4 // gain map - tile\_3

essential=1, property\_index=11;

essential=0, property\_index=10;  
 essential=0, property\_index=8;

essential=1, property\_index=9;

10) item\_ID=10, association\_count=4 // gain map - tile\_4

essential=1, property\_index=11;

essential=0, property\_index=10;  
 essential=0, property\_index=8;

essential=1, property\_index=9;

11) item\_ID=11, association\_count=5 // tmap

essential=0, property\_index=1;

essential=1, property\_index=12;

essential=0, property\_index=13;

essential=0, property\_index=14;

essential=1, property\_index=15;

ItemDataBox 'idat':

Grid derivation data block (at idat\_offset P1, with length Q1)

Grid derivation data block (at idat\_offset P6, with length Q6)

Tone-map derivation data block (at idat\_offset P11, with length Q11)

GroupsListBox 'grpl':

EntityToGroupBox 'altr': group\_id=12, num\_entities\_in\_group=2

entity\_id=11;

entity\_id=1;

MediaDataBox 'mdat':  
 HEVC Image (at file offset P2, with length Q2)

HEVC Image (at file offset P3, with length Q3)

HEVC Image (at file offset P4, with length Q4)

HEVC Image (at file offset P5, with length Q5)

HEVC Image (at file offset P7, with length Q7)

HEVC Image (at file offset P8, with length Q8)

HEVC Image (at file offset P9, with length Q9)

HEVC Image (at file offset P10, with length Q10)

# New colour format enhancement derived image item

Add the following new subclause after subclause 6.6.2.4:

## Colour format enhancement derivation

### Definition

An item with an item\_type value of 'cfen' is a colour format enhancement derived image item whose reconstructed image is formed from one or more input images that carry components in the luma plane that are used to reconstruct a picture with an enhanced colour format.

The input images are ordered using the reference type 'dimg' for this colour format enhancement derived image item within the ItemReferenceBox, where the value of from\_item\_ID identifies the colour format enhancement derived image item, and the values of to\_item\_ID identify the input images. The reference\_count gives the number of input image items and shall be greater or equal to one.

The colour format enhancement derived image item shall:

* have a Pixel Information property, an Image Spatial Extents property, and a Colour Information property with colour\_type set to 'nclx'.

EDITORS NOTE: Version 1 of 'pixi' property could be mandated for the use with 'cfen'.

The input image item(s):

* shall not signal duplicate entries for channel\_id greater than 1.
* shall each have a Pixel Information property and an Image Spatial Extents property.
* can have a Colour Information property that shall match the signaling from the Colour Information property of the colour formant enhancement derived image item.

EDITORS NOTE: It should be investigated if this restriction should only apply for version 0 of 'cfen'

The colour format enhancement derived image item should be signaled as a displayable item. The first entry signaled by the reference\_count array may be signaled as a displayable item. All remaining items signaled by the reference\_count array shall be hidden.

NOTE: Both the colour format enhancement derived image item and the first entry signaled by the reference\_count array can be incorporated into an 'altr' alternative group to signify that both image items are interlinked alternatives of each other. The colour format enhancement derived image item can be signaled as the first item in the 'altr' alternative group, while the second item in the group can store a 4:2:0 coded image to allow backwards compatibility.

### Syntax

aligned(8) class ColourFormatEnhancement {  
 unsigned int(8) version = 0;  
 for (i=0; i<reference\_count; i++) {  
 bit(7) reserved = 0;  
 unsigned int(1) is\_packed\_flag;  
 if(is\_packed\_flag == 1) {  
 // packed replacement of components  
 bit(2) reserved = 0;  
 unsigned int(3) num\_cols\_minus1;  
 unsigned int(3) num\_rows\_minus1;  
 unsigned int(5) hor\_guard\_band\_mul2;  
 unsigned int(5) ver\_guard\_band\_mul2;  
 for(m=0; m<num\_rows\_minus1+1; m++) {  
 for(n=0; n<num\_cols\_minus1+1; n++) {  
 unsigned int(8) packed\_channel\_id;  
 }  
 }  
 }  
 else {  
 unsigned int(8) channel\_id;  
 }  
 }  
}

### Semantics

version shall be equal to 0 for this version of specification.

is\_packed\_flag indicates a flag that determines if the referenced item contains multiple components packed into a single picture. When is\_packed\_flag is set to 1, the referenced item contains multiple components packed into the first channel, i.e. the luma channel. When is\_packed\_flag is set to 0 then the referenced item does not contain packed components in the first channel. When components are packed in the same item, such components all have the same horizontal and vertical resolution.

num\_cols\_minus1 plus 1 specifies the number of columns partitioning the picture.

num\_rows\_minus1 plus 1 specifies the number of rows partitioning the picture.

hor\_guard\_band\_mul2 specifies if a horizontal guard band region is present between two picture channels and, if present, its size. If hor\_guard\_band\_mul2 is equal to 0, then no horizontal guard band is present. If hor\_guard\_band\_mul2 is larger than 0, then the horizontal guard band between two picture channels is equal to hor\_guard\_band\_mul2\*2.

ver\_guard\_band\_mul2 specifies if a vertical guard band region is present between two picture channels and, if present, its size. If ver\_guard\_band\_mul2 is equal to 0, then no vertical guard band is present. If ver\_guard\_band\_mul2 is larger than 0, then the vertical guard band between two picture channels is equal to ver\_guard\_band\_mul2\*2.

channel\_id or packed\_channel\_id provide the identifier for the channel for the referenced items.

|  |  |  |
| --- | --- | --- |
| channel\_id or packed\_channel\_id | Mapping (depending on the 'colr' box) | |
| 0 | Unused | |
| 1 | Unspecified | |
| 2 | Y | R |
| 3 | Cb | G |
| 4 | Cr | B |
| 5 | Alpha | |
| 6 | Depth | |
| 7-255 | Reserved for future use. | |

# New constrained extents grid property

Add the following new subclause after subclause 6.5.40:

## Constrained Extents Grid Property

### Definition

|  |  |
| --- | --- |
| Box type: | 'cexg' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | No |
| Quantity (per item): | At most one |
|  |  |

The ConstrainedExtentsGridProperty descriptive item property indicates that each extent of the associated image item in the itemLocationBox is constrained to enclose data units of the item that are extractable as a contiguous byte range and are independently decodable and renderable as image tiles.

The configuration data needed to decode each extent independently may be present in the ExtentDecoderConfigurationRecord within the ConstrainedExtentsGridProperty. If the configuration data is not present in the ConstrainedExtentsGridProperty, all data units or properties required to configure the decoder and decode an image tile shall be declared in the decoder configuration and initialization properties associated with the image item.

The reconstructed image of the associated image item is formed from one or more image tiles in a given grid order within a larger canvas.

The image tiles corresponding to the extents are inserted in row-major order, top-row first, left to right, in the order of the extents for the associated image item within the ItemLocationBox. The value of extent\_count within the ItemLocationBox shall be equal to (1+rows\_minus\_one)\*(1+columns\_minus\_one). All image tiles shall have exactly the same width and height, image\_tile\_width and image\_tile\_height. The reconstructed image is formed by tiling the image tiles into a grid with a column width equal to image\_tile\_width and a row height equal to image\_tile\_height, without gap or overlap. The grid of image tiles shall completely “cover” the reconstructed image of the associated image item, where image\_tile\_width\*columns is greater than or equal to image\_width and image\_tile\_height\*rows is greater than or equal to image\_height, where image\_width and image\_height are signalled in the ImageSpatialExtentsProperty associated with the image item.

The flags field is used to signal image tiles related information. The following flags values are defined.

0x000001 field\_length\_flag: when set to 1 specifies that the length of the fields image\_tile\_width and image\_tile\_height is 32 bits. When field\_length\_flag is set to 0 specifies that the length of the fields image\_tile\_width and image\_tile\_height is 16 bits.

0x000002 tile\_config\_info\_present\_flag: when set to 1, specifies that the decoder configuration data needed to decode each extent independently is present in the ConstrainedExtentsGridProperty. When tile\_config\_info\_present\_flag is set to 0, all data units or properties required to configure the decoder and decode the extent independently shall be declared in the decoder configuration and initialization properties associated with the image item.

### Syntax

aligned(8) class ConstrainedExtentsGridProperty  
extends ItemFullProperty('cexg', version = 0, flags) {

// this is a temporary,non-parsable variable  
 unsigned int FieldLength = ((flags & field\_length\_flag) + 1) \* 16;  
 unsigned int(16) rows\_minus\_one;  
 unsigned int(16) columns\_minus\_one;  
 unsigned int(FieldLength ) image\_tile\_width;  
 unsigned int(FieldLength ) image\_tile\_height;

if(flags & tile\_config\_info\_present\_flag){

ExtentDecoderConfigurationRecord() extent\_config;

}  
}

### Semantics

image\_tile\_width, image\_tile\_height: specify respectively the width and height in pixels of the image tiles.

rows\_minus\_one, columns\_minus\_one: specify the number of rows of image tiles, and the number of image tiles per row. The value is one less than the number of rows or columns respectively. Image tiles enclosed in extents populate the top row first, followed by the second row and following rows, in the order of extents.

ExtentDecoderConfigurationRecord is the decoder configuration record needed to decode the corresponding extents. The decoder configuration record is specific and defined by the image coding format used for encoding the extent data.

# Stereo related signaling

## 5.1 Clarifications on camera extrinsic and intrinsic properties

In subclause 6.5.39.1, replace

The CameraExtrinsicMatrixProperty describes the spatial setup of camera(s). It specifies a position, in the cartesian representation, and an orientation, in the quaternion representation, of an orthogonal right-handed camera coordinate system within an orthogonal right-handed cartesian 3D world coordinate system.

NOTE 1 When needed, the information on the cameras GPS position can be provided using EXIF as defined in Annex A.

With

The CameraExtrinsicMatrixProperty describes the spatial setup of camera(s). It specifies a position, in the cartesian representation, and an orientation, in the quaternion representation, of an orthogonal right-handed camera coordinate system within an orthogonal right-handed cartesian 3D world coordinate system. The CameraExtrinsicMatrixProperty describes the camera extrinsics of the item output image.

NOTE 1 As specified in clause 6.5.1, all descriptive properties that come after transform properties are ignored. This means that CameraExtrinsicMatrixProperty needs to precede all transform properties even though it describes the output image.

NOTE 2 When needed, the information on the cameras GPS position can be provided using EXIF as defined in Annex A.

In subclause 6.5.39.1, renumber NOTE 2 to NOTE 3.

In subclause 6.5.40.1, replace

The CameraIntrinsicMatrixProperty describes the characteristics of the camera that captured the associated image item.

With

The CameraIntrinsicMatrixProperty describes the characteristics of the camera that captured the associated image item. The CameraIntrinsicMatrixProperty describes the camera intrinsics of the item output image.

NOTE 1 As specified in clause 6.5.1, all descriptive properties that come after transform properties are ignored. This means that CameraIntrinsicMatrixProperty needs to precede all transform properties even though it describes the output image.

In subclause 6.5.40.1, replace

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

*fx* = focal\_length\_x × image\_width / *denominator*

*fy* = focal\_length\_y × image\_height / *denominator*

*cx* = principal\_point\_x × image\_width / *denominator*

*cy* = 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.

With

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

*fx* = focal\_length\_x × *imageWidth* / *denominator*

*fy* = focal\_length\_y × *imageHeight* / *denominator*

*cx* = principal\_point\_x × *imageWidth* / *denominator*

*cy* = principal\_point\_y × *imageHeight* / *denominator*

*s* = skew\_factor / *skewDenominator*

where

*imageWidth* and *imageHeight* are the output dimensions of the image item after applying all transformative properties, if any. If no transformative properties are associated with the item this translates respectively to the dimensions image\_width and image\_height in the ImageSpatialExtentsProperty.

In subclause 6.5.40.1, renumber NOTE 2 to NOTE 3 and NOTE 3 to NOTE 4.

Add the following new subclause after subclause 6.5.41:

## Disparity adjustment information

### Definition

|  |  |
| --- | --- |
| Box type: | 'dadj' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per associated identifier value): | No |
| Quantity (per associated identifier value): | At most one |
|  |  |

The disparity adjustment descriptive item property defines the suggested global disparity adjustment amount for a stereo pair.

This item property should only be associated with an item or entity group that describes a stereo pair. If associated with a stereo pair entity group, the FileTypeBox or the ExtendedTypeBox associated with the FileTypeBox shall indicate that the requirements of the 'unif' brand apply in the file.

EDITORS NOTE: See TuC on frame-packed stereo items

### Syntax

aligned(8) class DisparityAdjustmentProperty  
extends ItemFullProperty('dadj', version = 0, flags = 0) {  
 signed int(32) disparity\_adjustment;  
}

### Semantics

disparity\_adjustment in units of 1/10 000 image widths. Positive values denote increased disparity with respect to parallel view direction.

Add the following new subclause after subclause 6.5.42:

## Stereo aggressors item property

### Definition

|  |  |
| --- | --- |
| Box type: | 'stag' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per associated identifier value): | No |
| Quantity (per associated identifier value): | Zero or more |
|  |  |

The stereo aggressors descriptive item property specifies the presence and characteristics of stereo aggressors detected within a stereo pair. Stereo aggressors are identified as elements that potentially cause discomfort when viewing the stereo pair on a stereoscopic display. This item property allows to identify and characterize these aggressors in detail.

This item property, when present, shall be used with a stereo pair. It can also be associated with other items only if also associated with a stereo pair so it is clear in which stereo context it applies. If associated with a stereo pair entity group ('ster' or 'stem'), the 'unif' brand shall be present in the compatible brands.

If this item property is associated with another item in addition to the stereo pair, it serves as a hint that the aggressors described by the property are localized to that specific item. For example, this can be used to indicate that a "Lens occlusion" is present in only the left or right image item, or using a region item, in a specific region of an image item. Multiple stereo aggressor properties may be associated with the same stereo pair since not all aggressors may be localized to the same area.

NOTE 1: An image item can be associated with multiple stereo pair entity groups; however a stereo aggressor can only apply to a specific group, which is why there is a requirement that the stereo aggressor has to be associated with a stereo group.

NOTE 2: As localization of the stereo aggressors is indicated with item property associations, all stereo aggressors within a single 'stag' box need to share the same localization. Specifying multiple unique localizations requires using multiple 'stag' boxes.

The aggressor\_severity shall be interpreted as follows:

* A value of 0 indicates an unknown severity. The file creator has no recommendation for the reader.
* A value of 1 to 42 (inclusive) indicates a mild severity. The file creator recommends that sensitive viewers are warned about potential discomfort when viewing in stereo.
* A value of 43 to 84 (inclusive) indicates a medium severity. The file creator recommends that all viewers are warned about potential discomfort when viewing in stereo.
* A value of 85 to 127 (inclusive) indicates a high severity. The file creator recommends that all viewers are warned about potential discomfort when viewed in stereo and that viewing should default to monoscopic rather than stereoscopic display unless overridden by the viewer.

If sub\_type\_uri is not specified for a specific aggressor, the aggressor\_severity is a relative measure only used to provide the viewer recommendations above and to rank the aggressors in the file relative to each other. If sub\_type\_uri is specified for an aggressor, aggressor\_severity shall be an absolute measure that can be compared between files.

### Syntax

aligned(8) class StereoAggressorsProperty  
extends ItemFullProperty('stag', version = 0, flags = 0) {  
 unsigned int(8) aggressor\_count\_minus\_one;  
 for(int i = 1; i <= aggressor\_count\_minus\_one + 1; i++) {  
 unsigned int(8) aggressor\_type;  
 unsigned int(1) sub\_type\_present;  
 unsigned int(7) aggressor\_severity;  
 if (sub\_type\_present) {  
 utf8string sub\_type\_uri;  
 }  
 }  
}

### Semantics

version shall be equal to 0.

aggressor\_count\_minus\_one is the number of aggressors minus one.

aggressor\_type has the following defined values:

|  |  |
| --- | --- |
| 0 | Unspecified |
| 1 | Lens occlusion |
| 2 | Image condition mismatch (lighting conditions, color, glare, sharpness, processing differences) |
| 3 | Stereo window violation |
| 4 | Object(s) too close |
| 5 | Stereo calibration error(s) |
| 6 | Temporal alignment mismatch |
| 7 | Poor image quality in one or both views (e.g. due to specific noise patterns) |
| 8 - 255 | Reserved |

aggressor\_severity provides an indication of how severe the file creator considers the aggressor to be. If sub\_type\_uri is not defined, the severity is file-relative and can not be used to compare aggressor severities between files.

sub\_type\_uri is an optional null-terminated UTF-8 character string of the Uniform Resource Identifier (URI) used to more explicitly identify the type of the stereo aggressor as well as how to interpret the aggressor\_severity. If specified, the aggressor\_severity shall be an absolute metric that can be compared between files. If not present, it defaults to the empty string.

Add the following text to the end of subclause 6.8.5:

NOTE: The 'stem' entity group can be used to indicate which item should be used for monoscopic fallback if the primary item is not part of the 'ster' group.

Add the following new subclause after 6.8.9:

## 'stem' entity grouping

The stereo pair entity grouping with monoscopic fallback ('stem'), is functionally equivalent to the 'ster' entity grouping with the following changes:

* The entity group shall contain exactly three entity\_id values that point to image items. The first two of these values shall be interpreted in the same way as the two entity\_id values of a 'ster' entity group.
* The third entity\_id value (with i equal to 2) indicates the image to display if monoscopic viewing of the stereo pair is desired. This may be the same as either the first or second entity\_id, but may also point to another image item more suitable for monoscopic display.
* If the primary item is part of the 'stem' group, the third entity\_id shall designate the primary item for monoscopic fallback.
* The value of (flags & 3) indicates how the third entity\_id relates to the stereo pair. The following values are defined:  
  0: Unspecified  
  1: Co-located with left view  
  2: Co-located with right view  
  3: Centered between left and right views

NOTE: To be backwards compatible with parsers that do not understand 'stem' entity groups, file creators may add both a 'stem' and a 'ster' entity group to the file. An 'altr' entity group can then be added containing the group\_id of the 'stem' and 'ster' entity group in that order to indicate that only one should be used. In order to use group\_id's in an 'altr' entity group, the 'unif' brand is required.

# New HDR signaling aligning with ISO 22028-5

Add the following new subclauses after subclause 6.5.43:

EDITORS NOTE: Do we need these properties for video as well?

## Reference viewing environment

### Definition

|  |  |
| --- | --- |
| Box type: | 'reve' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | No |
| Quantity (per item): | At most one |
|  |  |

The reference viewing environment applies to display-viewing colorimetry, not to scene-referred colorimetry. It specifies the luminance and chromaticity parameters for the “surround” and “periphery” of the display. The “surround” is the area surrounding a display that can affect the adaptation of the eye, typically the wall or curtain behind the display, while “periphery” is the remaining environment outside of the surround.

### Syntax

class ReferenceViewingEnvironmentBox extends ItemFullProperty('reve', 0, 0){  
 unsigned int(32) surround\_luminance;  
 unsigned int(16) surround\_light\_x;  
 unsigned int(16) surround\_light\_y;  
 unsigned int(32) periphery\_luminance;  
 unsigned int(16) periphery\_light\_x;  
 unsigned int(16) periphery\_light\_y;  
}

### Semantics

surround\_luminance specifies the luminance of the surround in units of 0.0001 candelas per square metre.

[Ed. note]: disallowing 0 could be considered.

surround\_light\_x and surround\_light\_y specify the normalized x and y chromaticity coordinates, respectively, of the environmental reference surround light in the nominal viewing environment. These parameters are according to the CIE 1931 definition of x and y as specified in ISO 11664-1 (see also ISO 11664-3 and CIE 15) and are in normalized increments of 0.0001. The values of surround\_light\_x and surround\_light\_y shall be in the range of 0 to 10000, inclusive.

periphery\_luminance specifies the luminance of the periphery in units of 0.0001 candelas per square metre.

[Ed. note]: disallowing 0 could be considered.

periphery\_light\_x and periphery\_light\_y specify the normalized x and y chromaticity coordinates, respectively, of the environmental reference periphery light in the nominal viewing environment. These parameters are according to the CIE 1931 definition of x and y as specified in ISO 11664-1 (see also ISO 11664-3 and CIE 15) and are in normalized increments of 0.0001. The values of periphery\_light\_x and periphery\_light\_y shall be in the range of 0 to 10 000, inclusive.

## Nominal Diffuse White

### Definition

|  |  |
| --- | --- |
| Box type: | 'ndwt' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | No |
| Quantity (per item): | At most one |
|  |  |

### Syntax

class NominalDiffuseWhiteBox extends ItemFullProperty('ndwt', 0, 0){  
 unsigned int(32) diffuse\_white\_luminance;  
}

### Semantics

diffuse\_white\_luminance indicates the default nominal diffuse white luminance in units of 0.0001 candelas per square metre. The chromaticity information associated with the diffused white luminance is the same as the chromaticity information specified in the ColourInformationBox of the associated image. If diffuse\_white\_luminance is set to 0 then the default definition of the ISO/TS 22028-5 should be used.

# Unified identifier handling clarifications

Replace the word "item\_ID" in clauses 6.5.18.1, 6.5.19.1, 6.5.20.1, 6.5.27.1, 6.5.35.1 with the phrase "identifier value"

Append the following text as a new paragraph in clauses 6.5.18.1, 6.5.19.1, 6.5.20.1, 6.5.27.1, 6.5.35.1:

If this property is associated with an entity group, the FileTypeBox or the ExtendedTypeBox associated with the FileTypeBox shall indicate that the requirements of the 'unif' brand apply in the file.

# Overview images

Add the following new subclause after subclause 6.4.9:

## Overview images

An overview image is described by a grid derived image item or a tiled pre-derived coded image item or a tiled image item whose reconstructed image is formed from generating a lower resolution, ‘binned’ version of the reconstructed image of a base image item. The base image item is also a tiled image item. The tiling may be implemented using a feature of a specific codec, or by using a grid derived image item, or by using a tiled image item. When a grid derived image item is used, the input items to the grid define the tiles. Derived image items shall not be used as inputs to the image grid, due to the need for in place byte range accessing of content. Individual tiles shall be written contiguously in memory, thereby allowing access with a single read or write action.

A pre-derived coded image item representing an overview image or an image item representing the base image that are tiled using a feature of a specific codec shall be stored in such a way that each extent identifies that data range corresponding to a tile and shall be associated with a ConstrainedExtentsGridProperty indicating the constraint on the extents and describing the tiling grid.

In cases where the binned resolution results in a fractional, or incomplete tile at the end of a row (column), the last tile in a row (column) of tiles shall be padded with the value zero at the end of the row (column) to complete the last tile in the row (column). If necessary, the clean aperture transformative property ('clap') may be applied to crop padded rows and/or columns. The number of tiles in a row (column) of tiles is determined by dividing the width (height) of the overview image by the tile size in X (tile size in Y) and rounding up.

The image format of the overview images is the same as the base image. i.e. number of bands, bit depth, color format, etc.

Overview images can be stacked together with the base image as a series of progressively binned images in an image pyramid entity group as defined in 6.8.11.

NOTE 1: In this version of the document, the exact derivation process (approaches such as the sum, average, median, minimum, or maximum value of a binned region) used to produce an overview from the base image is left to the implementer.

NOTE 2: When removing (or modifying) an item that is marked as the base image of an overview image, the content of associated image overview items might need to be removed (or rewritten).

Add the following new subclause after subclause 6.8.10:

## Image Pyramid Entity Group

### Definition

Box Type: 'pymd'

Container: GroupsListBox in a MetaBox at file level

Mandatory: No

Quantity: Zero or more

The ImagePyramidEntityGroup indicates a set of image items, formed as a base image item and a series of progressively binned overview image items, which together form the layers of an image pyramid.

The ImagePyramidEntityGroup provides overall information for the individual tiles inside the overview image items and base image item of the image pyramid.

The image format of the overview images shall be the same as the base image (i.e. number of bands, bit depth, color format, etc).

This entity group shall contain entity\_id values that point to a base image item and a set of overview image items and shall contain no entity\_id values that point to tracks. The entities shall be listed in the order of lowest resolution overview image item to the highest resolution overview image item, followed finally by the base image item of the image pyramid.

The flags field is used to signal image tiles related information. The following flags values are defined.

0x000001 tile\_info\_present\_flag: when set to 1, specifies that the tile information is present. When set to 0, the tile information is not present and is derived as described below.

0x000002 tile\_info\_constant\_flag: when set to 1, specifies that the tile size is constant across all the layers of the image pyramid. When set to 0, the tile size may not be constant across all the layers of the image pyramid.

When the flag tile\_info\_present\_flag is set, the tile information of a layer of the image pyramid is defined as follows:

*tileWidth* = tile\_size\_x

*tileHeight* = tile\_size\_y

*tileColumns* = ceil(*ispe*.image\_width / *tileWidth*)

*tileRows* = ceil(*ispe*.image\_height / *tileHeight*)

where *ispe* is the ImageSpatialExtentsProperty associated with the image item.

When the flag tile\_info\_present\_flag is not set, the tile information of a layer of the image pyramid is derived depending on the image item as described in Tables 2 to 5 below.

Rename the Table No’s in the integrated document.

**Table 2 — Tile information based on ConstrainedExtentsGridProperty associated with the image item**

|  |  |
| --- | --- |
| **ImagePyramidEntityGroup tile information** | **ConstrainedExtentsGridProperty** |
| *tileWidth* | image\_tile\_width |
| *tileHeight* | image\_tile\_height |
| *tileColumns* | (columns\_minus\_one)+1 |
| *tileRows* | (rows\_minus\_one)+1 |

**Table 3 — Tile information based on Tiled image item 'tili'**

|  |  |
| --- | --- |
| **ImagePyramidEntityGroup tile information** | **Tiled image item 'tili'** |
| *tileWidth* | tile\_width |
| *tileHeight* | tile\_height |
| *tileColumns* | ceil(*ispe*.image\_width/tile\_width) |
| *tileRows* | ceil(*ispe*.image\_height/tile\_height) |

**Table 4 — Tile information based on a grid derived image item with ImageGrid payload**

|  |  |
| --- | --- |
| **ImagePyramidEntityGroup tile information** | **ImageGrid** |
| *tileWidth* | ceil(output\_width/(columns\_minus\_one+1)) |
| *tileHeight* | ceil(output\_height/(rows\_minus\_one+1)) |
| *tileColumns* | columns\_minus\_one+1 |
| *tileRows* | rows\_minus\_one+1 |

**Table 5 — Tile information based on 'uncC' item property as defined in ISO/IEC 23001-17 associated with the image item**

|  |  |
| --- | --- |
| **ImagePyramidEntityGroup tile information** | **'uncC' item property** |
| *tileWidth* | ceil(*ispe*.image\_width/(num\_tile\_cols\_minus\_one+1)) |
| *tileHeight* | ceil(*ispe*.image\_height/(num\_tile\_rows\_minus\_one+1)) |
| *tileColumns* | num\_tile\_cols\_minus\_one+1 |
| *tileRows* | num\_tile\_rows\_minus\_one+1 |

There may be multiple ImagePyramidEntityGroups in the same file with different group\_id values.

NOTE All the entities of a same ImagePyramidEntityGroup, or only some of them, can also be members of a same entity group of type 'prgr' if they are stored in the file for allowing a progressive refinement. They can also be members of a same entity group of type 'altr' if they are proposed by the content creator as alternatives to be displayed for players not supporting the ImagePyramidEntityGroup.

When using region partition groups jointly with an image pyramid, the area covered by a region partition group should correspond to the area of a tile of the image pyramid.

A region item may be associated with an image item within an ImagePyramidEntityGroup either:

* through an item reference of type 'cdsc' from the region item to the image item;
* through a RegionPartitionGroupBox associated with the image item via an item reference of type 'rpds' and referencing the item\_ID of the region item.

A region item associated with a base image or an overview image within a same ImagePyramidEntityGroup may be applied to the output image of any image item within this ImagePyramidEntityGroup by applying the implicit resampling caused by the difference between the reference space of the region item and the size of the image.

NOTE A player can use the item reference of type 'base' of a merge region item to filter the region items that are inherited from other image items in the ImagePyramidEntityGroup.

### Syntax

aligned(8) class ImagePyramidEntityGroup  
extends EntityToGroupBox ('pymd', version = 0, flags) {  
 if(flags & tile\_info\_present\_flag){

unsigned int num\_of\_tile\_info; // this is a temporary,non-parsable variable

if(flags & tile\_info\_constant\_flag){

num\_of\_tile\_info = 1;

}else {

num\_of\_tile\_info = num\_entities\_in\_group;

}  
 for(i=0; i<num\_of\_tile\_info;i++) {  
 unsigned int(32) tile\_size\_x;  
 unsigned int(32) tile\_size\_y;  
 }

}

}

### Semantics

num\_entities\_in\_group as defined for EntityToGroupBox. In addition, it also specifies the number of layers of the image pyramid.

tile\_size\_x, tile\_size\_y indicate the size in pixels of a tile in the width and height dimension, respectively, for all layers of the image pyramid if num\_of\_tile\_info is equal to 1, or for each layer of the image pyramid in the order of the lowest resolution overview image item to the highest resolution overview image item if num\_of\_tile\_info is greater than 1.

# Tiled image item

Add the following new definition in subclause 3.1:

**3.1.XX tiled image item**

An image item constructed using uniform tiled subregions, allowing direct access to individual tiles.

Add the following new subclause after subclause 6.11:

## 6.12 Tiled image items

**6.12.1 General**

A tiled image item is constructed of uniform, independently coded tiles arranged in rows, columns, and optionally extra dimensions, to form a rectangular image or n-dimensional hyperrectangle. The tiles are identical in size, format, coding, and makeup and may be compressed or uncompressed.

**6.12.2 Tiled image item**

An image item of type ‘tili’ is a tiled image, with each tile coded independently from other tiles. Input tiles may be stored either in separate external files or in a contiguous range of addressing space to support byte range addressing and retrieval of individual tiles with a single read.

NOTE 1 The image coding method is defined by a writer using a valid image codec 4CC.

NOTE 2 As opposed to a ‘grid’ image item, where the declaration and addressing of tiles occurs in the file-scoped MetaBox, a ‘tili’ image item with contiguous range of addressing space has a single declaration parameter in the file-scoped MetaBox and an associated addressing table, with offsets and extents for each tile, stored with the image tiles, typically in a media data box. This has the advantage that the required file ranges of the addressing table, which can be large for terapixel images, can also be loaded on-demand.

The tiled image item (‘tili’) shall be associated with TiledImageConfigurationBox (‘tilC’) and ImageSpatialExtentsProperty which carries the width and height of the overall tiled image. All necessary configuration properties for the given image item type shall be defined in the ‘tilC’ property and stored in the tile\_image\_property[] array.

When the overall image dimensions are not an even multiple of the image tile size, the rows are padded on the right to complete the last tile in each row of tiles, and the columns are padded on the bottom to complete the last tile in each column of tiles. The width and height parameters in the ImageSpatialExtentsProperty are set to the size of the image containing valid image content, effectively achieving a crop of the padded boundary area.

**6.12.3 Tiled image configuration**

**6.12.3.1 Definition**

Box type: 'tilC'

Property type: Descriptive item property

Container: ItemPropertyContainerBox

Mandatory (per item): Yes, for image items of type 'tili'

Quantity (per item): One

The TiledImageConfigurationBox specifies parameters associated with a tiled image item (‘tili’) (see Section 6.12). These parameters include the tile resolution, and the image item type used to code and store individual tile content. Configuration information also includes the number and size of additional dimensions when coding n-dimensional hyperrectangles. This includes support for the coding of multi and hyperspectral imagery where each band in a tili tile region is separately retrievable.

**6.12.3.2 Syntax**

aligned(8) class TiledImageConfigurationBox

extends ItemFullProperty('tilC', version=0, flags) {

unsigned int(32) tile\_width;

unsigned int(32) tile\_height;

sequential\_order = ((flags>>4) & 0x01);

external\_tiles\_urls = ((flags>>5) & 0x01);

switch (flags & 0x03) {

case 0:

offset\_field\_length = 32;

break;

case 1:

offset\_field\_length = 40;

break;

case 2:

offset\_field\_length = 48;

break;

case 3:

offset\_field\_length = 64;

break;

}

switch ((flags>>2) & 0x03) {

case 0:

size\_field\_length = 0;

break;

case 1:

size\_field\_length = 24;

break;

case 2:

size\_field\_length = 32;

break;

case 3:

size\_field\_length = 64;

break;

}

unsigned int(8) number\_of\_extra\_dimensions;

for (int i=0; i<number\_of\_extra\_dimensions; i++) {

unsigned int(32) dimension\_size[i];

}

if (external\_tiles\_urls==0) {

unsigned int(32) tile\_item\_type;

unsigned int(8) number\_of\_tile\_properties;

for (int i=0; i<number\_of\_tile\_properties; i++) {

ItemProperty tile\_image\_property[i];

}

}

}

**6.12.3.3 Semantics**

tile\_width, tile\_height shall be set to the size of a single tile width and height. All tiles have the same size. Tiles at the right or bottom border of the overall image may include padding when the tile width and or height are not integer multiples of the overall tili item width or height. In this case, the ImageSpatialExtentsProperty is set to the boundary of the true image width and height to achieve a crop of the padded area.

external\_tiles\_urls when set to 0, specifies that the input tile images are each stored using the TiledImageOffsetTable and the tile related configuration information is present in TiledImageConfigurationBox

external\_tiles\_urls when set to 1, specifies that the input tile images are each stored in external files as indicated by the URLs in DataEntryTiledItemURLBox and the tile related configuration information is not present in TiledImageConfigurationBox. In this case, the tile image item data is empty.

[Ed. note]: we study the compression of URLs and the offset table to further optimize the design.

tile\_item\_type specifies the image item type used for all the individual tile images. In a tili item, each tile is coded separately so it can be extracted and decoded independently. tile\_item\_type shall be set to a valid four-character code for a coded image item (e.g., ‘hvc1’ for h265 compression, ‘j2k1’ for JPEG2000, or ‘unci’ for uncompressed). When required by the image item type, all necessary image properties shall be stored in tile\_image\_property[] Certain codecs (jpg, etc.) may not require any configuration properties.

number\_of\_extra\_dimensions specifies the number of extra dimensions if the image resembles a (number\_of\_extra\_dimensions+2)-dimensional hyperrectangle. For a 2D image, number\_of\_extra\_dimensions shall be 0.

dimension\_size[i] specifies the size of dimension i+2 of the n-dimensional hyperrectangle. Note that the size of the first two dimensions are the image\_width and image\_height specified in the ImageSpatialExtentsProperty of the tili item.

sequential\_order when true, indicates that the compressed image tile data is stored consecutively in sequential order.

number\_of\_tile\_properties specifies the number of tile properties stored in the TiledImageConfigurationBox.

tile\_image\_property[] are the image item properties used when decoding a tile image. This includes at least all mandatory item properties for an image item of type tile\_item\_type with the exception of ImageSpatialExtentsProperty. If tile\_image\_property[] does not contain an ImageSpatialExtentsProperty, the decoder shall synthesize an ImageSpatialExtentsProperty with tile\_width and tile\_height as the size.

[Ed. note]: all the item properties needed for tiles are to be studied.

offset\_field\_length defines the number of bits used to store the offset to the image data of a specific tile in the TiledImageOffsetTable.

size\_field\_length defines the number of bits used to store the length of the image data of a specific tile in the TiledImageOffsetTable.

**6.12.4 Tiled image item data**

The payload of a tiled image item (‘tili’) consists of a TiledImageOffsetTable and the tiles of the item when the external\_tiles\_urls is set to 0.

The TiledImageOffsetTable contains offset pointers and size information for each tile in the item. This table is then followed by the set of coded image tile data.

aligned(8) class TiledImageOffsetTable{

for (int i=0; i < NumTiles; i++) {

unsigned int(OffsetFieldLength) tile\_start\_offset[i];

unsigned int(SizeFieldLength) tile\_size[i]; //not present if SizeFieldLength==0

}

// *... followed by coded tile data ...*

tile\_start\_offset[i] points to the start of the coded data of a tile. The position is given relative to the start of the TiledImageOffsetTable. If a specific tile is empty and does not contain image content, the tile is not coded and the tile\_start\_offset[i] entry shall be set to 0. This situation may occur when an image is generated on a canvas and certain portions of the overall image only contain canvas with no image pixels. Readers shall interpret a tile\_start\_offset[i] value equal to 0 as an empty tile with no media content. Note, a tile\_start\_offset[i] value is not a file offset, but an offset into the item's data that can potentially span several iloc extents.

tile\_size[i] (if present) indicates the number of bytes of the coded tile bitstream.

The number of tile offsets stored in the table (NumTiles) is computed by

TileColumns = ceil(ImageSpatialExtentsProperty.image\_width/tile\_width);

TileRows = ceil(ImageSpatialExtentsProperty.image\_height/tile\_height);

NumTiles = TileColumns \* TileRows

for (i=0; i<number\_of\_extra\_dimensions; i++) {

NumTiles = NumTiles \* dimension\_size[i];

}

TileColumns and TileRows are the number of tiles in a row within the overall image and the number of tiles in a column within the overall image. image\_width and image\_height are the dimensions of the entire image as specified in the mandatory ImageSpatialExtentsProperty item property. number\_of\_extra\_dimensions and dimension\_size[] is defined in the TiledImageConfigurationBox property associated with the tili item. NumTiles represents the number of tiles in the entire tiled image item.

The entries in the offset table are ordered in row-major sequence. For a 2D image with a single coded layer, they are indexed as [y][x], where:

x = tile column

y = tile row

For a 3D tiled image item, they are indexed as [z][y][x], where:

x = tile column

y = tile row

z = depth coordinate

For a general n-dimensional hyperrectangle, the tiles are indexed as [zn-1] [zn-2] ...[z3] [z2][y][x], where zi are the n-2 extra dimensions.

x is the inner most looping variable, followed by y, and then z2 to zn-1.

The coded tile data may be stored in the file in arbitrary order, resulting in the tile\_start\_offset entries not necessarily being in increasing address order.

When size\_field\_length==0, the tile\_size[i] variables are not present, and the decoder infers them from the difference between the tile\_start\_offset entries. For the case where tiles are stored in sequential order (flags & 0x10 == 0x10), the tile\_size[i] is computed as

tile\_start\_offset[i+1] - tile\_start\_offset[i], except for the last tile, which extends until the end of the data. If the tiles are not stored in sequential order, the decoder first sorts the tile start offsets before computing the size from the offset differences. In this case, the decoder cannot read the offset table on-demand. For on-demand applications, the tile sizes should be included. When multiple tiles contain the same content, the tile\_start\_offset entries for these tiles may point to the same data block. In this case, sequential ordering is not used.

* + 1. **Data entry tiled item URL box**

#### 6.12.4.1 Definition

Box Type: 'deti'  
Container: DataReferenceBox  
Mandatory: No.  
Quantity: Zero or more.

The DataEntryTiledItemURLBox identifies the location of the external files which carry the input image items of a tiled image item ('tili'). Each location identified by the DataEntryTiledItemURLBox corresponds to an input image item with a specific item\_ID.

#### 6.12.4.2 Syntax

unsigned int(8) function f(unsigned int(3) index) {  
 switch(index) {  
 case 0: return 8;  
 case 1: return 16;  
 case 2: return 24;  
 case 3: return 32;

case 4: return 40;  
 case 5: return 48;  
 case 6: return 56;  
 case 7: return 64;  
}

aligned(8) class DataEntryTiledItemURLBox (bit(24) flags)  
 extends DataEntryBaseBox('deti', flags)   
{

unsigned int(3) input\_items\_size\_index;

bit(5) reserved = 0;  
 unsigned int(f(input\_items\_size\_index)) no\_of\_input\_items;

for(i=0;i<no\_of\_input\_items;i++){

utf8string location;

}  
}

#### 6.12.4.3 Semantics

input\_items\_size\_index specifies the size of the parameters no\_of\_input\_items in bytes. With value 0 indicating size is of 1 byte up to the value 7 indicating the size to be 8 bytes.

The parameter no\_of\_input\_items in DataEntryTiledItemURLBox shall be equal to:

TileColumns = (ImageSpatialExtentsProperty.image\_width + TiledImageConfigurationBox.tile\_width-1)/ TiledImageConfigurationBox.tile\_width;

TileRows = (ImageSpatialExtentsProperty.image\_height + TiledImageConfigurationBox.tile\_height-1)/ TiledImageConfigurationBox.tile\_height;

no\_of\_input\_items = TileColumns \* TileRows

for (i=0; i<number\_of\_extra\_dimensions; i++) {

no\_of\_input\_items = no\_of\_input\_items \* dimension\_size[i];

}

location indicates the location of the referred file as a URL. The URL can be an absolute or a relative URL, and the located resource shall be a compliant HEIF file. Relative URLs are relative to the file that contains this location.

# Region partition group

Add the following new subclause after subclause 6.8.11:

## Region Partition Group

### Definition

Box Type: 'rgpa'

Container: GroupsListBox in a MetaBox at file level

Mandatory: No  
Quantity: Zero or more

A region partition group lists region items that are contained inside an area of an image. This area is the area covered by the region partition group and is defined as a rectangle in the image.

The region partition group is associated with the image item containing the region items it groups through an item reference of type 'rpds' from the image item to the region partition group.

Depending on flags values, the area covered by a region partition group is either defined in the region partition group structure itself or is the area of the image item the region partition group is associated with through an item reference of type 'rpds'.

The following flags values are defined to describe the area covered by the region items contained in a region partition group:

0x001000, large\_field\_length; when set specifies that the length of the fields reference\_width, reference\_height, top, left, width, and height is 32 bits. When not set, it specifies that the length of the fields reference\_width, reference\_height, top, left, width, and height is 16 bits.

0x002000, area\_info\_present; when set, specifies that the fields reference\_width, reference\_height, top, left, width, and height are present. When not set, it specifies that the fields reference\_width, reference\_height, top, left, width, and height are not present. In this case, the area covered by the region partition group is the area of the image it is associated with.

A diagram of a diagram

Description automatically generated with medium confidence

Figure 1: Example of two region partition groups ('rgpa') associated with an image, the area of the top group is the whole image, while the area of the bottom group is the bottom-center part of the image

A region partition group associated with an image item should only contain region items associated with the image item or with another image item that this image item is a part of.

A diagram of a grid

Description automatically generated

Figure 2: Example of two region partition groups ('rgpa') related to a grid, the top group is associated with the grid and its area is the whole image, the bottom group is associated with an input item of the grid and its area is the whole input item

When a region item is contained in a region partition group, at least one of its regions shall intersects the area covered by the region partition group.

A region partition group shall list only region items it contains. A region partition group may list all the region items it contains. A region partition group may list only the region items it contains and that are adapted to being rendered when the rendered size of the area covered by the region partition group is greater than or equal to the display area.

When rendering a part of an image item, a renderer may use region partition groups to select which region items to parse and render. The renderer may use region partition groups matching the area to display. When several region partition groups correspond to the area to render, the renderer may use those matching the scale of the rendered area. The renderer may use region partition groups associated with the image item to render or associated with items corresponding to parts of the image item to render.

### Syntax

aligned(8) class RegionPartitionGroupBox  
extends EntityToGroupBox('rgpa', version, flags) {  
 unsigned int field\_size = ((flags & large\_field\_length)?32:16);  
 if (flags & area\_info\_present) {  
 unsigned int(field\_size) reference\_width;  
 unsigned int(field\_size) reference\_height;  
 unsigned int(field\_size) top;  
 unsigned int(field\_size) left;  
 unsigned int(field\_size) width;  
 unsigned int(field\_size) height;  
 }  
}

### Semantics

reference\_width, reference\_height specify, in pixel units, the width and height, respectively, of the reference space in which the region partition group is specified.

top, left specify the coordinates of the top-left corner of the area covered by the region partition group relatively to the reference space.

width, height specify the coordinates of the width and the height of the area covered by the region partition group relatively to the reference space.

# Change all mention of SingleItemTypeReferenceBox to include SingleItemTypeReferenceBoxLarge

EDITORS NOTE: Once we have the text for HEIF 3ed, make sure that we have not missed any mention of SingleItemTypeReferenceBox.

In subclause 6.6.1 replace:

The number of SingleItemTypeReferenceBoxes with the box type 'dimg' and with the same value of from\_item\_ID shall not be greater than 1.

With:

The number of SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge entries with the box type 'dimg' and with the same value of from\_item\_ID shall not be greater than 1.

In subclause 6.6.2.2.1 replace:

The input images are listed in the order they are layered, i.e. the bottom-most input image first and the top-most input image last, in the SingleItemTypeReferenceBox of type 'dimg' for this derived image item within the ItemReferenceBox.

With:

The input images are listed in the order they are layered, i.e. the bottom-most input image first and the top-most input image last, in the SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge of type 'dimg' for this derived image item within the ItemReferenceBox.

In subclause 6.6.2.2.3 replace:

reference\_count is obtained from the SingleItemTypeReferenceBox of type 'dimg' where this item is identified by the from\_item\_ID field.

With:

reference\_count is obtained from the SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge of type 'dimg' where this item is identified by the from\_item\_ID field.

In subclause 6.6.2.3.1 replace:

The input images are inserted in row-major order, top-row first, left to right, in the order of SingleItemTypeReferenceBox of type 'dimg' for this derived image item within the ItemReferenceBox. In the SingleItemTypeReferenceBox of type 'dimg', the value of from\_item\_ID identifies the derived image item of type 'grid', the value of reference\_count shall be equal to rows\*columns, and the values of to\_item\_ID identify the input images.

With:

The input images are inserted in row-major order, top-row first, left to right, in the order of SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge of type 'dimg' for this derived image item within the ItemReferenceBox. In the SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge of type 'dimg', the value of from\_item\_ID identifies the derived image item of type 'grid', the value of reference\_count shall be equal to rows\*columns, and the values of to\_item\_ID identify the input images.

In subclause 6.11.1 replace:

The number of SingleItemTypeReferenceBoxes with the box type 'drgn' and with the same value of from\_item\_ID shall not be greater than 1.

With:

The number of SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge entries with the box type 'drgn' and with the same value of from\_item\_ID shall not be greater than 1.

# Add T.35 metadata to Annex A

In subclause A.1 replace:

This annex specifies the format to store metadata complying with Exif (JEITA CP-3451), XMP (ISO 16684-1), MPEG-7 (ISO/IEC 15938-3), or IPTC Information Interchange Model (IPTC-IIM) in files conforming to the Image File Format. When Exif, XMP, MPEG-7, or IPTC-IIM metadata is associated with items or tracks conforming to the Image File Format, the metadata shall follow the specifications of this annex. However, it is not required for a reader conforming to this document to understand Exif, XMP, MPEG-7, or IPTC-IIM metadata.

With:

This annex specifies the format to store metadata complying with Exif (JEITA CP-3451), XMP (ISO 16684-1), ITU-T T.35, MPEG-7 (ISO/IEC 15938-3), or IPTC Information Interchange Model (IPTC-IIM) in files conforming to the Image File Format. When Exif, XMP, ITU-T T.35, MPEG-7, or IPTC-IIM metadata is associated with items or tracks conforming to the Image File Format, the metadata shall follow the specifications of this annex. However, it is not required for a reader conforming to this document to understand Exif, XMP, ITU-T T.35, MPEG-7, or IPTC-IIM metadata.

Add the following after subclause A.5:

## A.6 ITU-T T.35 metadata

ITU-T T.35 metadata as specified may be associated with image items using an item type of 'it35' as defined in ISO/IEC 14496-12.

For image sequences, ITU-T T.35 metadata may be embedded within samples and signalled using an ITU-T T.35 sample group or may be stored in an ITU-T T.35 sample group as defined in ISO/IEC 14496-12.

Add the following in the Bibliography:

*[xx]* Recommendation ITU-T T.35*, Procedure for the allocation of ITU-T defined codes for non-standard facilities*

# Clarify that items may share auxiliary images, thumbnails and metadata

In subclause 6.4.4 replace:

The thumbnail image and the master image are linked using a reference type 'thmb' from the thumbnail image to the master image.

With:

The thumbnail image and master image(s) are linked using a reference type 'thmb' from the thumbnail image to the master image.

In subclause 6.4.5 replace:

The auxiliary image and the master image are linked using an item reference of 'auxl' from the auxiliary image to the master image.

With:

The auxiliary image and master image(s) are linked using an item reference of 'auxl' from the auxiliary image to the master image.

# Merge region item

*Add the following definition in 3.1 (Terms and definitions)*

**3.1.XX**

**Merge region item**

*region item* (3.1.44) associated with an *image item* (3.1.18) that represents an approximate, merge or summary of one or more *region items* (3.1.44) associated with one or more other *image items* (3.1.18)

*Add a new subclause in 11.3 (Regions and region annotations for an image item)*

**11.3.4 Merge region item**

If a region item associated with an image item represents an approximate, merge or summary of region items associated with other image items then it shall be linked to those region items by item references of type 'base' from the region item to region items of other image items, it approximates, merges or summarizes. A region item including a 'base' item reference is referred to as a merge region item.

NOTE 1 Determining that a region item for an image is an approximate, merge or summary of region items of another image is deliberately left to the file creator.

A merge region item may be associated with region annotations. The region annotations associated with region items referenced by a merge region item through a 'base' item reference also apply to this merge region item.

NOTE 2 A player can follow the 'base' item references of a merge region item to retrieve and possibly render the region annotations associated with the referred region items as part of the region annotations associated with this merge region item.