ISO/IEC JTC 1/SC 29/WG 03 N0985

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Technology under Consideration on ISO/IEC 23008-12

**Status:** Approved

**Date of document:** 2023-08-04

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 25 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0985**

**July 2023 – Geneva, CH**

|  |  |
| --- | --- |
| **Title** | **Technology under Consideration on ISO/IEC 23008-12** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22983** |

Abstract

This document collects following candidate technologies for the High Efficiency Image File Format (HEIF) (ISO/IEC 23008-12).

Table of Contents

[1 Region annotations for image sequence or video tracks 3](#_Toc142013271)

[1.1 Region extrapolation (from m60304, MPEG#139, Issue#76) 3](#_Toc142013274)

[1.1.1 Overview 3](#_Toc142013275)

[1.1.2 Text Proposal 3](#_Toc142013276)

[1.2 Region interpolation (from m59508, MPEG#138, Issue#69 comment#60556) 6](#_Toc142013277)

[1.2.1 Text Proposal 6](#_Toc142013278)

[1.2.2 Discussion 7](#_Toc142013279)

[2 Region annotation for image items 7](#_Toc142013280)

[2.1 Combination of regions (from m62028, MPEG#141, Issue#88) 7](#_Toc142013281)

[2.1.1 Summary/Motivation 7](#_Toc142013282)

[2.1.2 Text Proposals 8](#_Toc142013283)

[3 Matrix-based transformation for image items 9](#_Toc142013284)

[4 Signaling for pre-derived coded image items 9](#_Toc142013285)

[5 On HDR signaling alignment with ISO 22028-5 (from m62055, MPEG#141, Issue#86) 9](#_Toc142013286)

[5.1 Motivation 9](#_Toc142013287)

[5.2 Text proposal 10](#_Toc142013288)

[6 On MPEG/JPEG file embedding (MPEG#141, Issue#87) 12](#_Toc142013289)

[6.1 Discussion 12](#_Toc142013290)

[6.2 Initial text proposal 12](#_Toc142013291)

[7 Non-droppable predicted frames (MPEG#142, Issue#90) 12](#_Toc142013292)

[7.1 Proposal 12](#_Toc142013293)

[8 Image Overviews in HEIF (MPEG#143, Issue#102) 13](#_Toc142013294)

[8.1 Introduction 13](#_Toc142013295)

[8.2 Use cases 13](#_Toc142013296)

[8.3 Requirements 13](#_Toc142013297)

[8.4 Implementation Approach 14](#_Toc142013307)

[8.5 Proposal 14](#_Toc142013308)

[8.6 Draft Implementation 14](#_Toc142013309)

[9 Extending pixi for more use-cases (MPEG#143, Issue#95) 16](#_Toc142013313)

[9.1 Pixel information 17](#_Toc142013314)

[9.1.1 Definition 17](#_Toc142013315)

[9.1.2 Syntax 17](#_Toc142013316)

[9.1.3 Semantics 17](#_Toc142013317)

[10 Tone map derivation item for gain map HDR (MPEG#143, Issue#101) 17](#_Toc142013318)

[10.1 Gain map 18](#_Toc142013319)

[10.1.1 Gain map data 18](#_Toc142013320)

[10.1.2 Gain map metadata 18](#_Toc142013321)

[10.2 HEIF storage 19](#_Toc142013322)

[10.3 Tone-map derivation 22](#_Toc142013324)

[10.3.1 Definition 22](#_Toc142013325)

[10.3.2 Syntax 24](#_Toc142013326)

[10.3.3 Semantics 24](#_Toc142013327)

# Region annotations for image sequence or video tracks

## Region extrapolation (from [m60304](https://dms.mpeg.expert/doc_end_user/documents/139_OnLine/wg11/m60304-v1-m60304-Regionextrapolationfortracks.zip), MPEG#139, [Issue#76](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/76))

### Overview

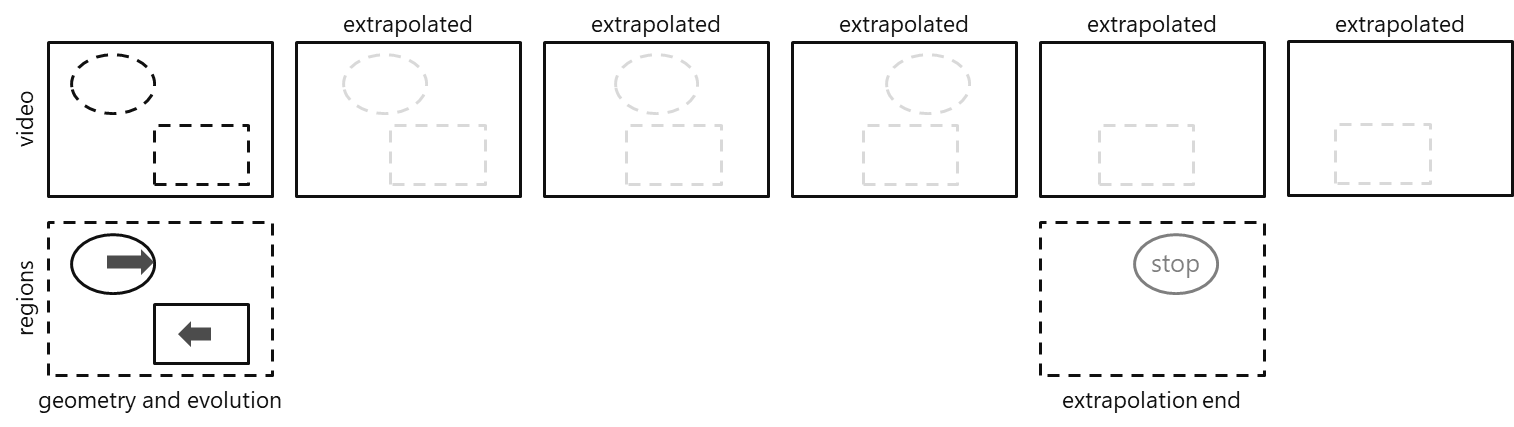


Figure 1: example region description for tracks using extrapolation

Figure 1 shows an example of describing several regions using extrapolation. The video track shown at the top contains two regions, an elliptic one and a rectangular one. The elliptic one is present in the four first samples of the video track and moves to the right of the image. The rectangular one is present in all the samples of the video track and moves to the left of the image.

The region track shown at the bottom describes these two regions. In a first sample, corresponding to the first sample of the video track, these two regions are described with their positions and sizes and the evolution of their respective positions and sizes. There are no region samples corresponding to the three following video samples. The region sample corresponding to the fifth video sample signals that the interpolation of the elliptic region ends.

### Text Proposal

*Update the definition of a region track ( section 7.5.4.1) by adding the following paragraphs:*

The geometry of a region may be defined by specifying the shape, position and size of the region in a sample of the region track. The geometry of a region may also be defined as an initial geometry and its evolution over time by specifying the initial geometry of the region and its evolution in a sample of the region track.

The evolution of a region over time is optional. It can be represented by the evolution speed of some of its parameters inside the reference space. The evolution speed of the parameters is signaled using a scaling factor for increasing its precision. The parameters defining the evolution of a region depend on the geometry of the region as follows:

— When the geometry of a region is represented by a point, the evolution of the region is defined by the evolution of the position of this point.

— When the geometry of a region is represented by a rectangle or an ellipse, the evolution of the region is defined by the evolution of the position and the size of the rectangle or ellipse.

— When the geometry of a region is represented by a polygon or a polyline, the evolution of the region is defined by the evolution of the position of each point of the polygon or polyline. The number of points in the polygon or polyline doesn’t change.

— When the geometry of a region is represented by a mask, the evolution of the region is defined by the evolution of the position of the mask.

The evolution of a region stops when another sample contains a region with the same region identifier. The evolution of a region shall stop for each sync sample of the source track.

*Update the Sample format (section 7.5.4.2.1) with the following paragraph*

When the extrapolate flag is set to 1 for a region inside a sample of a region track, the region is an evolving region defined by an initial geometry and its evolution over time.

The value of each evolving parameter defining the geometry of the region at a given composition time *T* can be computed as follows:

where:

* *param0* is the initial value of the parameter as defined in the initial geometry of the region at time T0.
* *Δparam* is the evolution of the parameter as defined in the evolution of the region.
* *evolution\_scale* is a scaling factor for the evolution values equal to , where is the field\_size and is equal to ((RegionTrackConfigBox.field\_length\_size & 1) + 1) \* 16.
* *T0* is the composition time of the sample defining the evolving region.
* *ΔT* is the duration of the sample defining the evolving region.

*Update the syntax of Sample format (section 7.5.4.2.2) as follows*

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;  
 unsigned int(1) extrapolate;  
 unsigned int(7) reserved;  
 if (geometry\_type == 0) {  
 // point  
 signed int(field\_size) x;  
 signed int(field\_size) y;  
 if (extrapolate == 1) {  
 signed int(field\_size) delta\_x;  
 signed int(field\_size) delta\_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;  
 if (extrapolate == 1) {  
 signed int(field\_size) delta\_x;  
 signed int(field\_size) delta\_y;  
 signed int(field\_size) delta\_width;  
 signed int(field\_size) delta\_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;  
 if (extrapolate == 1) {  
 signed int(field\_size) delta\_x;  
 signed int(field\_size) delta\_y;  
 signed int(field\_size) delta\_radius\_x;  
 signed int(field\_size) delta\_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;  
 }  
 if (extrapolate == 1) {  
 for (i=0; i < point\_count; i++) {  
 signed int(field\_size) delta\_px;  
 signed int(field\_size) delta\_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;  
 if (extrapolate == 1) {  
 signed int(field\_size) delta\_x;  
 signed int(field\_size) delta\_y;  
 }  
 }  
 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[];  
 if (extrapolate == 1) {  
 signed int(field\_size) delta\_x;  
 signed int(field\_size) delta\_y;  
 }  
 }  
 else if (geometry\_type == 7) {  
 // empty region  
 }  
 }  
}

*Update the semantics of Sample format (section 7.5.4.2.3) with the following text:*

7: the region is an empty region used for signalling the end of the evolution of a previous region with the same region identifier.

Other values are reserved.

extrapolate is a flag indicating whether the geometry changes of the region are specified or not. When equal to 0, it indicates that no geometry changes are specified for the region. When equal to 1, it indicates that both the geometry and the geometry changes are specified for the region.

(…)

evolution\_scale is the scaling factor for the specification of the evolution values, equal to , where is the field\_size and is equal to ((RegionTrackConfigBox.field\_length\_size & 1) + 1) \* 16.

delta\_x, delta\_y specify, in 1/evolution\_scale units of the reference space, the evolution of the x and y fields for the region.

delta\_width, delta\_height specify, in 1/evolution\_scale units of the reference space the evolution of the width and height fields for the region.

delta\_radius\_x, delta\_radius\_y specify, in 1/evolution\_scale units of the reference space the evolution of the radius\_x and radius\_y fields for the region.

delta\_px, delta\_py specify, in 1/evolution\_scale units of the reference space the evolution of the px, py fields for a point of the region.

## Region interpolation (from [m59508](https://dms.mpeg.expert/doc_end_user/documents/138_OnLine/wg11/m59508-v1-m59508-Regionannotationfortracks.zip), MPEG#138, [Issue#69 comment#60556](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/69" \l "note_60556))

*[Ed. (FD)This section only contains parts of the contribution requiring further discussion, i.e., the interpolate flag in sample format for region tracks]*

### Text Proposal

**X.X Region track and region annotations for an image sequence or video track**

**X.X.3 Sample format**

**X.X.3.1 Definition**

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

**X.X.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(7)reserved;  
 unsigned int(1)interpolate;  
 unsigned int(16) region\_count;  
 for (r=0; r < region\_count; r++) {  
 (…)  
 }  
}

**X.X.3.3 Semantics**

interpolate indicates the continuity in time of the successive samples. When true, the application may linearly interpolate values of the region geometries between the previous sample and the current sample. When false, there shall not be any interpolation of values between the previous and the current samples.

NOTE 1 When using interpolation, it is expected that the interpolated samples match the presentation time of the samples in the referenced source track. For instance, for each video sample of a video track, one interpolated region sample is calculated.

(…)

### Discussion

About the interpolate flag: The purpose is to avoid declaring a sample in the region track for each sample of the media track when regions are moving linearly between two positions. Imagine a sample A in the region track with a region at a starting position A and this region is moving linearly to the arrival position B nine samples later. Instead of declaring ten samples in the region track, you can only declare two samples, sample A with a duration corresponding to nine samples in the media track, followed by sample B providing the arrival position B. We should clarify that since the interpolate flag applies to all regions in the sample, the number of regions shall be the same in sample A and B.

# Region annotation for image items

## Combination of regions (from [m62028](https://dms.mpeg.expert/doc_end_user/documents/141_OnLine/wg11/m62028-v1-m62028-Regioncombination.zip), MPEG#141, [Issue#88](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/88))

### Summary/Motivation

This contribution proposes two extensions for region annotation feature:

1. The ability to associate annotations to a region defined as the union of several regions

A new type of derived region item (items 100 and 101 in Figure 1) is defined to allow grouping several regions to signal that those regions are part of a group and to be able to annotate this group as a whole (e.g. these persons are part of the same family)

1. The ability to indicate a relation between several regions

A new type of entityToGroup is defined to allow signaling that a region item (person in figure 2 below) represents/covers an area including several other regions (as an aggregation) (left arm, right leg, etc...). In terms of geometry, the inclusion does not need to be total.

|  |  |
| --- | --- |
|  |  |
| Figure 1 - Example of union of regions | Figure 2 – Example of inclusion relationship |

### Text Proposals

#### Proposal 1: Union of regions

*Add the following section in section 6.11 Derived region items*

**6.11.2.2 Union derivation**

An item with an item\_type value of 'cbrg' defines a derived region item that corresponds to the union of all the regions represented by one or more input region items.

The input region items are specified in a SingleItemTypeReferenceBox of type 'drgn' for this derived region item within the ItemReferenceBox. In the SingleItemTypeReferenceBox of type 'drgn', the value of from\_item\_ID identifies the derived region item of type 'cbrg' and the values of to\_item\_ID identify the input region items.

The union derived region item is associated with the image item inside which the regions are defined using an item reference of type 'cdsc' from the union derived region item to the image item.

The region resulting from this derived region item is the union of all the regions of each input region item after being applied to the referenced image item as specified in 6.10.1.1.

#### Proposal 2: relations between region items

*Add the following section in section 6.10 Region items and region annotations*

**6.10.4 Region Entity Group**

**6.10.4.1** 'corg' **Entity Group**

A compound region entity group ('corg') associates one main region item with one or more region items. It indicates an inclusion relationship between a main object covered by regions of a main entity and other objects covered by regions described by one or more other entities, the main object logically including the other objects.

NOTE For example, a compound region entity group can be used to associate a main region corresponding to a car with regions corresponding to wheels to indicate that the car is logically including the wheels.

The entities in a compound region entity group shall be region items. The number of entities in a compound region entity group shall be at least 2. The first entity\_id value shall indicate the main region item. It indicates the region covering the main object that is logically including the objects covered by the regions described by the second and following entity\_ids.

This inclusion relationship does not convey information at the geometry level. A main region signalled as including others regions by a compound entity group may or may not geometrically include the other regions.

*[[ Ed. (FM): MPEG#141: it was questioned:*

* *Can the Figure 1 in the proposal be achieved for example using mask items where all the regions belonging to a group is part of the mask item.*
* *Is there a restriction for any of the proposals that the separate regions must all be derived from a single image item?”]]*

# Matrix-based transformation for image items

*[[ Ed. (FD): MPEG#129: it was questioned:”* Should we also add ‘matrix’ as an image derivation in the HEIF? “. It was warned that “We would need to be clear about the meaning of outputs that don’t have horizontal and vertical sides; if that’s overlaid, the meaning is clear, but what if it’s supposed to be displayed?”*]]*

# Signaling for pre-derived coded image items

*Replace the clause 6.4.7 with the following text:*

**6.4.7** **Pre-derived coded images**

[Ed. (FD): In the following, differences with HEIF 2nd edition (w18310) are highlighted in blue]

If a coded image has been derived from others — for example, a composite HDR image derived from exposure-bracketed individual images, or a panorama derived from a set of images — then it shall be linked to those images by item references of type 'base'. Item references may be from the coded image to all images it derives from, or when unique IDs are used, from the coded image to all entity groups or images it derives from. When unique IDs are used, a to\_item\_ID value in the SingleItemTypeReferenceBox or SingleItemTypeReferenceBoxLarge is resolved to an item identifier whenever the embedding MetaBox contains an item with such identifier, and is resolved to an entity group identifier otherwise.

An image item including a 'base' item reference is referred to as a pre-derived coded image.

NOTE In this version of this document, the exact derivation process used to produce the image is not described.

[[Ed. (FD): At MPEG#129, it was commented that “The slight snag here is defining what it means when the entity group does NOT imply a single output (e.g. a slide show); what does pre-derivation mean? ]]

*Add the following clause as section 6.4.7.1:*

**6.4.7.1 Signaling of the derivation method for pre-derived coded image items**

A pre-derived coded image shall be linked to images it derives from by an item reference of type 'base' to the entity group containing all images the pre-derived coded images derives from. The grouping\_type of the EntityToGroupBox specifies the purpose of grouping and implicitly signals the type of the derivation operation which was applied to generate the pre-derived coded image.

[[Ed. (FM): At MPEG#126, it was commented that “we somehow need to indicate the derivation operation, rather than the nature of the input set”]]

[[Ed. (FD): At MPEG#129, it was commented that “We could allow a pre-derivation of the implied derivation of that entity group.”]]

# On HDR signaling alignment with ISO 22028-5 (from [m62055](https://dms.mpeg.expert/doc_end_user/current_document.php?id=85840), MPEG#141, [Issue#86](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/86))

## Motivation

ISO TC42 is developing specification ISO 22028-5 [1], which defines a set of colour image encodings for use in storage, transmission, and display of HDR and WCG digital still images. The purpose of this proposal is to initiate the alignment process with the definitions from the ISO 22028-5 and to provide the signaling for the related metadata at MPEG.

**Reference viewing environment metadata**

To properly interpret the color appearance of images encoded in a color image encoding, ISO 22028-1 specifies the reference image viewing environment that can be used to provide context for interpreting the intended color appearance of the encoded image colorimetry.

For cases where the image is to be viewed in an actual viewing environment significantly different than the specified reference image viewing environment, it might be desirable to use a colour appearance transform to determine corresponding image colorimetry that would produce the intended colour appearance in the actual viewing environment.

ISO 22028-5 [1] specifies parameters to establish a reference viewing environment in which images with display viewing colorimetry are intended to be viewed. These parameters include luminance of surround and periphery as well as color temperature of both.

**Nominal diffuse white luminance**

This data might be used by tone mapping algorithms at display time, to compensate differences between the reference display and the display of the user.

**Additional metadata**

Additional metadata for handling new transfer characteristics will also need to be defined as indicated in [2].

1. ISO TC 42/WG 23, TS 22028-5 "Photography and graphic technology — Extended colour encodings for digital image storage, manipulation and interchange — Part 5: High dynamic range and wide colour gamut image encoding for still images (HDR/WCG)"
2. N. Bonnier, D. Concion, D. Podborski, J. Roland, A. Tourapis: "ISO 22028-5 impact on CICP", [m62260](https://dms.mpeg.expert/doc_end_user/current_document.php?id=86045&id_meeting=193)

## Text proposal

**X..Y Reference viewing environment**

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.

**X.Y.1 Syntax**

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

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;  
}

**X.Y.2 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 10 000, 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.

**X.Z Nominal Diffuse White**

**X.Z.1 Syntax**

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

class NominalDiffuseWhiteBox extends ItemFullProperty('ndwt', 0, 0){  
 unsigned int(32) diffuse\_white\_luminance;  
 unsigned int(16) diffuse\_white\_light\_x;  
 unsigned int(16) diffuse\_white\_light\_y;  
}

**X.Z.2 Semantics**

diffuse\_white\_luminance indicates the default nominal diffuse white luminance in units of 0.0001 candelas per square metre.

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

diffuse\_white\_light\_x and diffuse\_white\_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 and should be the same as the chromaticity coordinates of the white point of the content.

[Ed. note]: We could also add a note mentioning that values diffuse\_white\_light\_x = 3 127 and diffuse\_white\_light\_y = 3 290 could be used to signal D65.

# On MPEG/JPEG file embedding (MPEG#141, [Issue#87](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/87))

## Discussion

During MPEG 140 (cf. [ISOBMFF/Issue#146](https://mpeg.expert/software/MPEG/Systems/FileFormat/isobmff/-/issues/146)), the potential improvement of ISOBMFF 8th edition was extended with a definition of the UUID (see text in section 6.2 below) to enable embedding an ISO base media file within another file. One of these use-cases would be to embed ISOBMFF in JPEG based on JUMBF ISO/IEC 19566-5, which would also allow HEIF files to be embedded into a JPEG file.

At MPEG#141, it was decided to remove the proposed text from ISOBMFF 8th edition for further study in HEIF. It was pointed out that embedding HEIF into JPEG may lead to sub-optimal encapsulation and compatibility issues. Uses cases were also questioned.

## Initial text proposal

*[Ed.(FM): The text below was initially included into potential improvement of ISOBMFF 8th edition clause 6.8 at MPEG#140 and then removed at MPEG#141 for further study]*

**6.8 UUID value for embedded ISO base media files**

When embedding an ISO base media file into a file compliant to another file format that needs a UUID to identify the format of the embedded file, the UUID to identify the ISO base media file shall be equal to 0x49534F30-0011-0010-8000-00AA00389B71.

NOTE This UUID enables embedding an ISO base media file within a file conforming to the JPEG Universal Metadata Box Format (JUMBF, ISO/IEC 19566-5). The JUMBF Content Type in the JUMBF Description box is set equal to the UUID specified above in this subclause. The JUMBF superbox contains a single content box that contains the ISO base media file.

# Non-droppable predicted frames (MPEG#142, [Issue#90](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/90))

Section B.3.2 in the HEIF specification prohibits the use of non-droppable predicted frames in HEIC sequences. This leaves a lot of coding gains off the table for sequences meant to be used as animations. At MPEG #142 it was proposed to relax this requirement into a recommendation.

The group agreed to allow relaxing of the rule, and add a note in the ccst box. The recommendation to use I frames every X seconds is expected to be further restricted in derived specifications and should not appear in HEIF specification.

## Proposal

Relax text in Section B.3.2:

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.

The suggested modification can be applied to the relevant sections for AVC, VVC, and EVC. Alternatively, consider relocating the text to a general section to prevent repetition. Another option is to remove this text, along with the associated paragraphs for AVC, VVC, and EVC, completely.

Add a note to Section 7.2.3.4:

all\_ref\_pics\_intra: This flag when set to one indicates the restriction that samples that are not sync samples, if any, are predicted only from sync samples.

NOTE 1 When there are inter predicted images in the track and all\_ref\_pics\_intra is equal to 1, then these images are all predicted from intra coded images.

NOTE2 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 may suggest guidelines for the frequency of sync samples.

# Image Overviews in HEIF (MPEG#143, [Issue#102](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/102))

## Introduction

The ability to efficiently access large still images over a network connection benefits from specific arrangement of pixel content within an image. Gridding within HEIF allows specific regions with pixels in close proximity to be accessed via efficient packaging, addressing and delivery. A similar capability is available through the tiling of uncompressed images via ISO/IEC 23001-17. This supports downloading sections of an image with manageable amounts of content appropriate for display on a given display size, whether it be a small phone device or a large 4K display. This approach avoids the need for having to download an entire image first before displaying content. When panning across a large image space, additional tiles are downloaded when the panning action reaches an additional tile. Tiles are cached on the receiving machine to facilitate efficient revisits to the areas already downloaded.

Graphical user interface

Description automatically generated with medium confidence

Figure 1‑1: Implementation of overviews in HEIF

While panning is supported by tiling, the ability to zoom and pan requires access to the imagery in a multi-resolution manner. To view the full field-of-view of a large image requires binning to squeeze the image onto a smaller pixel canvas. The creation of a multi-resolution image pyramid, or set of “overviews”, allows for efficient navigation via both pan and zoom operations.

This contribution proposes the creation of a pre-derived coded image type to support the implementation of overviews with HEIF imagery as well as an entity group to hold a base image and set of overviews, and a descriptive item property to facilitate communication of addressing information for individual tiles throughput the entire pyramid.

## Use cases

Applications accessing large imagery files over a network interface benefit from the implementation of multi-resolution overviews. This technique supports byte range addressing through a browser interface. Scanning satellite imagery in a wide variety of applications, ranging from mapping, to hiking and fitness apps, finding the nearest coffee shop, to real-estate apps benefit.

## Requirements

Requirements for the implementation of overviews and a pyramid within HEIF include:

* The ability to create tiled images, using either codec defined tiling or image grids
* The ability to choose the resolution of the common tile size for all images in a pyramid. i.e. 512 x 512, 1024 x 1024, etc.
* The ability to work with any image item format, regardless of color format, number of components, dynamic range, or pixel format
* A mechanism allowing region items (‘rgan’) to scale properly across the overview levels. Provide a mechanism to report this information for a region item at each level.
* Provide referencing and properties, as necessary, to label the features and relationships between all the multi-resolution images to facilitate remote indexing and byte range addressing in a straight-forward manner.
* An annotation mechanism that scales with the overviews.

## Implementation Approach

* Overviews are generated as pre-derived coded images, to facilitate byte range addressing to access any content of interest with minimal processing activity.
* Allow ‘grid’ derived images to support the tiling function of an overview. Because of the need for byte range addressing, the source image item defines the tiling for a grid.
* Allow codec specific tiling to support the implementation of tiling in overviews.
* Generate a series of multi-resolution, scaled images, with consistent tiling at each layer, and covering all the desired resolution scales.
* Leverage references and properties as needed to generate a full and efficient solution.
* Of note, for JPEG 2000 and HTJ2K, because of how the underlying wavelet compression works, imagery can be extracted from the main base image item with varying resolution. As a result, they use a different mechanism than described in this contribution.

## Proposal

* Upon generation of a satisfactory approach with group consensus, complete the design, document the approach, and submit for inclusion into amendment to 23008-12.

## Draft Implementation

**6.4.10 Image overviews**

**6.4.10.1 Definition**

An overview is a pre-defined coded image item whose reconstructed image is formed from generating a lower resolution, ‘binned’ version of a base image item. For overviews, the base item is a tiled image item. The tiling may be implemented using a feature of a specific codec, or by using a gridded image item. When a gridded 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.

The overview item shall be tiled using the same tiling scheme as the base image. i.e. if tiles in the base image are X by Y pixels, they are X by Y pixels in the overview. 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 values of 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.

An image pyramid is generated by stacking a series of progressively binned overviews and creating an entity group of type ‘pymd’. Each overview is referenced to the original full resolution base image, using a reference of type ‘base’ and each overview indicates its amount of binning by its overview level. An essential property of type ‘pmdp’ carries details on the storage location of the internal tiles for the overviews and base image making up a full pyramid. This enables simple query and quick navigation and access to specific tiles of interest in the set of overviews. The image format of the overviews is the same as the base image item. i.e. number of bands, bit depth, color format, etc.

Region items associated with the base image may be replicated for individual overviews using an appropriate scaling associated with the level of binning for a particular overview and referenced to the specific overview.

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 rewritten.

**6.5.37 Image Pyramid Information**

**6.5.37.1 Definition**

Box type: 'pmdp'

Property type: Descriptive item property

Container: ItemPropertyContainerBox

Mandatory (per associated item\_ID): Yes, for a pyramid entity group

Quantity (per associated item\_ID): At most one

The ImagePyramidInformation descriptive item property provides a manifest of addressing information for the individual tiles inside the overviews and base image item of an image pyramid. Image items are listed in the ordering given in the image pyramid entity group, which is from the lowest resolution overview to the base image. Tiles and addresses are listed in row major order. Addressing of tiles is zero based, with the first tile at (0,0)

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

The 'pmdp' item property, shall be associated to an image pyramid entity group.

**6.5.37.2 Syntax**

aligned(8) class ImagePyramidInformationProperty

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

unsigned int(8) num\_layers;

unsigned int(16) tile\_size\_x;

unsigned int(16) tile\_size\_y;

for(i=1;i<=num\_layers;i++){

unsigned int(8) layer\_num = i;

unsigned int(32) item\_ID;

unsigned int(8) layer\_binning;

unsigned int(16) tiles\_in\_layer\_row;

unsigned int(16) tiles\_in\_layer\_column;

for (j=0;j<tiles\_in\_layer\_row;j++){

for(k=0;k<tiles\_in\_layer\_column;k++){

unsigned int(16) row\_tile\_num=j;

unsigned int(16) column\_tile\_num=k;

unsigned int(64) tile\_offset;

unsigned int(32) tile\_byte\_count;

}

}

}

}

**6.5.37.3 Semantics**

num\_layers: Signals the number of overviews plus the layer for the base image.

tile\_size\_x: the size of a tile in the width dimension for all levels of the pyramid.

tile\_size\_y: the size of a tile in the height dimension for all levels of the pyramid.

layer\_num: Signals a layer, starting at ‘1’ for the lowest resolution overview, associated with the following tile information.

item\_ID: The item\_ID for the image at the indicated layer of the pyramid.

layer\_binning: Specifies the level of binning between the base image and the overview in the indicated layer of the pyramid. A 2x2 binning is defined to be a layer\_binning of 2, a 4x4 binning is defined to be 4, etc. The width and height for an overview with layer\_binning of 2 is half the width and half the height of the base image, etc. A base image has a layer\_binning of 1.

tiles\_in\_layer\_row: Signals the number of tiles in a row of a specific layer.

tiles\_in\_layer\_column: Signals the number of tiles in a column of a specific layer.

row\_tile\_num: The current row index for the tile information that follows

column\_tile\_num: The current column index for the tile information that follows

tile\_offset: Indicates the location of a specific tile in a specific layer

tile\_byte\_count: Indicates the size of a stored tile in a specific layer

**6.8.10 Image Pyramid Entity Group**

The image pyramid entity group ('pymd') indicates a set of entities, formed as a base image and a series of progressively binned overview images, which together form an image pyramid. Each overview is referenced to the original full resolution base image, using a reference of type ‘base’. An essential property of type ‘pmdp’ carries details on the set of images and the storage location of the internal tiles for the overviews and base image making up a full pyramid. This enables simple query and quick navigation and access to specific tiles of interest in the set of overviews. The image format of the overviews is the same as the base image item. i.e. number of bands, bit depth, color format, etc.

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

There may be multiple image pyramid entity groups in the same file with different group\_id values. An ImagePyramidInformationProperty shall be associated with the image pyramid entity group, and it shall be marked as essential.

# Extending pixi for more use-cases (MPEG#143, [Issue#95](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/95))

The initial proposal to extend the pixel information property 'pixi' was introduced in [m63650](https://dms.mpeg.expert/doc_end_user/current_document.php?id=87994) and is included in this section. During MPEG#143 it was discussed that improved signaling methods could be introduced, such as using box versioning instead of flags. In addition, the group is encouraged to make further suggestions on how 'pixi' can be extended to cover signaling for floating point, alpha (pre-multiplied or not), signed integers, phase shift for subsampled content, etc.

## Pixel information

### Definition

|  |  |
| --- | --- |
| Box type: | 'pixi' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | No, unless the premultiplied alpha is present in the associated image |
| Quantity (per item): | At most one |
|  |  |

The PixelInformationProperty descriptive item property indicates the number and bit depth of colour and alpha components, if present, in the reconstructed image of the associated image item.

The following flags are allowed to be set in the px\_flags only if version 2 is used:

0x000001 alpha-present: indicates that the coded item contains both color/monochrome pixels *and* alpha.

0x000002 premultiplied-alpha: indicates that the colour/monochrome pixels are premultiplied with the alpha. If this flag is set, the PixelInformationProperty box *shall* be marked essential. This flag shall be 0 unless alpha‑present is non-zero. If this flag is used, alpha‑present shall be set.

0x000004 is-floating-point: indicates that the channels are IEEE 754 floating-point values rather than integers. When this flag is set, bits\_per\_channel shall only take values 16, 32, 64, 128 or 256.

### Syntax

aligned(8) class PixelInformationProperty  
extends ItemFullProperty('pixi', version, px\_flags){  
 if (version == 0 || version == 2) {  
 unsigned int(8) num\_channels;  
 for (i=0; i<num\_channels; i++) {  
 unsigned int(8) bits\_per\_channel;  
 }  
 }  
 else if (version == 1) {  
 unsigned int(8) num\_channels;  
 for (i=0; i<num\_channels; i++) {  
 unsigned int(1) channel\_is\_signed;  
 unsigned int(7) bits\_per\_channel;  
 }  
 }  
}

### Semantics

num\_channels: This field signals the number of channels by each pixel of the reconstructed image of the associated image item.

bits\_per\_channel: This field indicates the bits per channel for the pixels of the reconstructed image of the associated image item. The permitted values for this field depend on the value of the is-floating-point flag.

# Tone map derivation item for gain map HDR (MPEG#143, [Issue#101](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/101))

There is ongoing work to standardize the concept of tone-mapping (and potentially colour-mapping) between two renditions of an image, facilitated by the introduction of a “gain map”. The concept has been proposed as a fresh work item under the ISO's Technical Committee 42 (TC 42), and has been officially accepted as new work item ISO/AWI 21496 1. Adobe has also referenced this concept in one of its white papers with a suggestion to store the gain map as an auxiliary image and the gain map metadata as XMP in HEIF. However, this signalling approach leads to a lot of ambiguities and should not be used in HEIF as there are better ways of signaling this content. This section includes a new HEIF-specific signaling that allow file creators to better express exactly how they want files with gain maps to be parsed and displayed.

## Gain map

### Gain map data

The current proposal on how to generate the gain map takes two renditions and computes the logarithm of the ratio of each color channel in linear RGB space. The gain-map may optionally be downscaled or reduced to a single component to save space. An affine transform is used to remap this ratio to the range from 0.0 to 1.0, inclusive. An optional custom gamma curve is applied, after which the content is converted to integer values and compressed with a supported codec.

A collage of mountains

Description automatically generated

Figure 1: Gain map generation process

From the point of view of the codec or container, the gain map data:

* is 1 or 3 channels of integers.
* uses undefined colour primaries (CICP value 2), or potentially the primaries used when computing the ratio assuming that both the base rendition and the alternate rendition have the same colour primaries.
* may use a CICP defined transfer function or may use the undefined transfer function (CICP value 2). If a custom gamma curve is used, it will be signaled in the gain map metadata.
* may use a YCbCr matrix and chroma subsampling to achieve better compression.

### Gain map metadata

The [Adobe white paper](https://helpx.adobe.com/content/dam/help/en/camera-raw/using/gain-map/jcr_content/root/content/flex/items/position/position-par/table/row-3u03dx0-column-4a63daf/download_section/download-1/Gain_Map_1_0d12.pdf) lists the following metadata as being needed to dynamically apply the gain map to the base rendition:

* Version
* BaseRenditionIsHDR (Boolean)
* HDRCapacityMin (float)
* HDRCapacityMax (float)
* GainMapMin[numChannels] (float)
* GainMapMax[numChannels] (float)
* Gamma[numChannels] (float)
* OffsetSDR[numChannels] (float)
* OffsetHDR[numChannels] (float)

The names and exact meaning of these fields may change as the ISO work item is finalized. We can generalize this as saying that we need to be able to store the following in HEIF:

* A version.
* Potentially some Boolean flags.
* Signaling floating point values as rational numbers (see section 5.3 for more details).
* Floating point/rational values that are not per channel.
* Arrays of 1 or 3 floating point/rational values for values that apply to each channel of the gain map.

## HEIF storage

Instead of using an auxiliary image as suggested in the [Adobe white paper](https://helpx.adobe.com/content/dam/help/en/camera-raw/using/gain-map/jcr_content/root/content/flex/items/position/position-par/table/row-3u03dx0-column-4a63daf/download_section/download-1/Gain_Map_1_0d12.pdf), it was proposed to create a new dedicated derived item type, which we will refer to as a **tmap** derived item. A HEIF file using this concept would consist of the following parts:

1. The base rendition image item (referred to as item 1).
2. The gain map data stored as a hidden image item (referred to as item 2).
3. A tmap derived item referencing items 1 and 2 (referred to as item 3).
4. An altr entity group that signals that the items 1 and 3 are alternative renditions of each other.

The gain map metadata is stored in the box body of item 3 (the tmap item) and therefore does not need any extra items.

Below is an example file layout for a file with an HDR base rendition, a 3-channel 10-bit gain map, and regular Exif and XMP metadata attached to the main image item:

('ftyp' "File Type Box") {...}  
('meta' "Meta Box") {  
 Version: 0, Flags: 0x000000  
 ('hdlr' "Handler Reference Box") {...}  
 ('pitm' "Primary Item Box") {  
 Version: 0, Flags: 0x000000  
 Item ID: 1  
 }  
 ('iinf' "Item Information Box") {  
 Version: 0, Flags: 0x000000  
 Entry count: 5  
 ('infe' "Item Info Entry") {  
 Item ID: 1  
 Item protection index: 0  
 Item type: 'hvc1'  
 Item name: "Base rendition"  
 }  
 ('infe' "Item Info Entry") {  
 Item ID: 2  
 Item protection index: 0  
 Item type: 'Exif'  
 Item name: "Exif metadata"  
 }  
 ('infe' "Item Info Entry") {  
 Item ID: 3  
 Item protection index: 0  
 Item type: 'mime'  
 Item name: "XMP metadata"  
 Content type: application/rdf+xml  
 Content encoding: None  
 }  
 ('infe' "Item Info Entry") {  
 Item ID: 4 (Hidden)  
 Item protection index: 0  
 Item type: 'hvc1'  
 Item name: "Gain Map image"  
 }  
 ('infe' "Item Info Entry") {  
 Item ID: 6  
 Item protection index: 0  
 Item type: 'tmap'  
 Item name: "Tone-mapped representation"  
 }  
 }  
 ('iloc' "Item Location Box") {  
 Version: 1, Flags: 0x000000  
 Offset size: 4  
 Length size: 4  
 Base offset size: 0  
 Index size: 0  
 Item count: 5  
 Item ID: 1  
 Construction method: 0, Data reference index: 0  
 Extent 1/1: Offset 757, Length 1712907  
 Item ID: 2  
 Construction method: 0, Data reference index: 0  
 Extent 1/1: Offset 1713664, Length 796  
 Item ID: 3  
 Construction method: 0, Data reference index: 0  
 Extent 1/1: Offset 1714460, Length 28276  
 Item ID: 4  
 Construction method: 0, Data reference index: 0  
 Extent 1/1: Offset 1742736, Length 752121  
 Item ID: 6  
 Construction method: 1, Data reference index: 0  
 Extent 1/1: Offset 0, Length 70  
 }  
 ('iprp' "Item Properties Box") {  
 ('ipco' "Item Property Container Box") {  
 ('ispe' "Image Spatial Extents") {  
 Version: 0, Flags: 0x000000  
 Image Width: 2400  
 Image Height: 3000  
 }  
 ('pixi' "Pixel Information Box") {  
 Version: 0, Flags: 0x000000  
 Bits per Channel: {10, 10, 10}  
 }  
 ('hvcC' "HEVC Decoder Configuration Record") {...}  
 ('colr' "Colour Information Box") {  
 colour\_type: 'nclx'  
 colour\_primaries: 9  
 transfer\_characteristics: 16  
 matrix\_coefficients: 9  
 full\_range\_flag: 1  
 }  
 ('clli' "Content Light Level Box") {  
 max\_content\_light\_level: 1459  
 max\_pic\_average\_light\_level: 75  
 }  
 ('ispe' "Image Spatial Extents") {  
 Version: 0, Flags: 0x000000  
 Image Width: 1200  
 Image Height: 1500  
 }  
 ('hvcC' "HEVC Decoder Configuration Record") {...}  
 ('colr' "Colour Information Box") {  
 colour\_type: 'nclx'  
 colour\_primaries: 1  
 transfer\_characteristics: 13  
 matrix\_coefficients: 6  
 full\_range\_flag: 1  
 }  
 ('colr' "Colour Information Box") {  
 colour\_type: 'nclx'  
 colour\_primaries: 2  
 transfer\_characteristics: 2  
 matrix\_coefficients: 6  
 full\_range\_flag: 1  
 }  
 }  
 ('ipma' "Item Property Association") {  
 Version: 0, Flags: 0x000000  
 Entry count: 3  
 Item ID: 1  
 Association count: 5  
 Property Index: 1, Essential: No  
 Property Index: 2, Essential: No  
 Property Index: 3, Essential: Yes  
 Property Index: 4, Essential: Yes  
 Property Index: 5, Essential: No  
 Item ID: 4  
 Association count: 3  
 Property Index: 6, Essential: No  
 Property Index: 2, Essential: No  
 Property Index: 7, Essential: Yes  
 Property Index: 9, Essential: Yes  
 Item ID: 6  
 Association count: 3  
 Property Index: 1, Essential: Yes  
 Property Index: 2, Essential: No  
 Property Index: 8, Essential: Yes  
 }  
 }  
 ('iref' "Item Reference Box") {  
 Version: 0, Flags: 0x000000  
 Reference type: 'cdsc' ( Content Describes (Metadata) )  
 From item ID: 2  
 Reference count: 1  
 To item ID: 1  
 Reference type: 'cdsc' ( Content Describes (Metadata) )  
 From item ID: 3  
 Reference count: 1  
 To item ID: 1  
 Reference type: 'dimg' ( Derived Image )  
 From item ID: 6  
 Reference count: 2  
 To item ID: 1  
 To item ID: 4  
 }  
 ('idat' "Item Data Box") {  
 Assumed tmap box data at offset 0:  
 Version: 0, Flags: 3  
 Components: 3; Base is HDR: True  
 HDRCapacityMin: 0/1  
 HDRCapacityMax: 28/10  
 GainMapMin: (0/1, 0/1, 0/1)  
 GainMapMax: (275377/100000, 2751731/1000000, 2743773/1000000)  
 Gamma: (311087/1000000, 317914/1000000, 286524/1000000)  
 OffsetSDR: (1/64, 1/64, 1/64)  
 OffsetHDR: (1/64, 1/64, 1/64)  
 }  
 ('altr' "Alternate Group Box") {  
 Version: 0, Flags: 0x000000  
 group id: 7  
 num entities in group: 2  
 entity id 1  
 entity id 6  
 }  
}  
('mdat' "Media Data Box") {...}

## Tone-map derivation

### 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 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 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 is considered as the base input image and the second as the gain map input image. The gain map input image may have different dimensions than the base as documented in ISO/AWI 21496‑1.

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

The base input image shall be associated with a colr item property.

The gain map input image shall be associated with a colr item property of type nclx. This property documents how to reverse any colour transformations that the encoder may have done for coding gains. The colour\_primaries field shall be set to 2 unless both renditions use the same colour primaries, in which case it may be set to that value. The transfer\_function field shall only differ from 2 if Gamma in the tmap box body is set to 1.0.

A tmap derived item shall be associated with a colr item property. This property describes the colour properties of the reconstructed image if the gain map input item is fully applied according to ISO/AWI 21496‑1.

[Ed note: The ISO/AWI 21496-1 will guide the definition of the base colour space, the other rendition colour space, and the working colour space. Subsequently, the colr boxes referenced here will be associated with these defined terms.]

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

NOTE 1: The gain map input image should be marked as not displayable by setting (flags & 1) equal to 0 in its infe entry.

NOTE 2: Backwards compatibility with parsers that do not support the tmap derivation may be achieved by placing the main input image item and the tmap derivation item in an altr entity group.

### Syntax

aligned(8) class ToneMapImage {  
 unsigned int(8) version = 0;  
 if(version == 0) {  
 unsigned int(8) flags; // 1 or 3  
 int ChannelCount = (flags & 1)\*2 + 1; // temp/nonparsable variable  
 Boolean BaseIsHDR = (flags & 2) != 0; // temp/nonparsable variable  
 unsigned int(32) HDRCapacityMinNumerator;  
 unsigned int(32) HDRCapacityMinDenominator;  
 unsigned int(32) HDRCapacityMaxNumerator;  
 unsigned int(32) HDRCapacityMaxDenominator;  
 unsigned int(32) GainMapMinNumerator [ChannelCount];  
 unsigned int(32) GainMapMinDenominator [ChannelCount];  
 unsigned int(32) GainMapMaxNumerator [ChannelCount];  
 unsigned int(32) GainMapMaxDenominator [ChannelCount];  
 unsigned int(32) GammaNumerator [ChannelCount];  
 unsigned int(32) GammaDenominator [ChannelCount];  
 unsigned int(32) HDROffsetNumerator [ChannelCount];  
 unsigned int(32) HDROffsetDenominator [ChannelCount];  
 unsigned int(32) SDROffsetNumerator [ChannelCount];  
 unsigned int(32) SDROffsetDenominator [ChannelCount];  
 }  
}

### Semantics

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

(flags & 1) equal to 1 specifies that the channel count of the gain map input is 3, while a value of 0 specifies that the channel count of the gain map input is 1.

(flags & 2) equal to 2 specifies that the base input image is the HDR rendition according to ISO/AWI 21496‑1.

HDRCapacityMin in ISO/AWI 21496‑1 is expressed as HDRCapacityMinNumerator/HDRCapacityMinDenominator.  
HDRCapacityMinDenominator shall not be 0.

HDRCapacityMax in ISO/AWI 21496‑1 is expressed as HDRCapacityMaxNumerator/HDRCapacityMaxDenominator.  
HDRCapacityMaxDenominator shall not be 0.

GainMapMin in ISO/AWI 21496‑1 is expressed per-channel as GainMapMinNumerator[i]/GainMapMinDenominator[i].  
GainMapMinDenominator[i] shall not be 0.

GainMapMax in ISO/AWI 21496‑1 is expressed per-channel as GainMapMaxNumerator[i]/GainMapMaxDenominator[i].  
GainMapMaxDenominator[i] shall not be 0.

Gamma in ISO/AWI 21496‑1 is expressed per-channel as GammaNumerator[i]/GammaDenominator[i].  
GammaDenominator[i] shall not be 0.

HDROffset in ISO/AWI 21496‑1 is expressed per-channel as HDROffsetNumerator[i]/HDROffsetDenominator[i].  
HDROffsetDenominator[i] shall not be 0.

SDROffset in ISO/AWI 21496‑1 is expressed per-channel as SDROffsetNumerator[i]/SDROffsetDenominator[i].  
SDROffsetDenominator[i] shall not be 0.