ISO/IEC JTC 1/SC 29/WG 03 N1137

**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:** 2024-02-09

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

**No. of pages:** 20 (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 N1137**

**January 2024 – Online**

|  |  |
| --- | --- |
| **Title** | **Technology under Consideration on ISO/IEC 23008-12** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **23504** |

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](#_Toc158289975)

[1.1 Region extrapolation (from m60304, MPEG#139, Issue#76) 3](#_Toc158289976)

[1.1.1 Overview 3](#_Toc158289977)

[1.1.2 Text Proposal 3](#_Toc158289978)

[1.2 Region interpolation (from m59508, MPEG#138, Issue#69 comment#60556) 6](#_Toc158289979)

[1.2.1 Text Proposal 6](#_Toc158289980)

[1.2.2 Discussion 7](#_Toc158289981)

[2 Region annotation for image items (from m62028, MPEG#141, Issue#88 and MPEG #145, issue #128) 7](#_Toc158289982)

[2.1 Discussion 7](#_Toc158289983)

*[2.1.1](#_Toc158289984)**[Motivation, use cases and initial proposal](#_Toc158289984)* [7](#_Toc158289984)

*[2.1.2](#_Toc158289985)**[Responses to open questions](#_Toc158289985)* [9](#_Toc158289985)

[2.2 Proposal 10](#_Toc158289986)

[2.2.1 Proposal 1: Union of regions 10](#_Toc158289987)

[11.3.3.2.2 Union derivation 10](#_Toc158289988)

[2.2.2 Proposal 2: Relations between region items 11](#_Toc158289989)

[11.3.4 Region Entity Group 11](#_Toc158289990)

[11.3.4.1 ‘corg’ Entity Group 11](#_Toc158289991)

[3 Matrix-based transformation for image items 11](#_Toc158290025)

[4 Signaling for pre-derived coded image items 11](#_Toc158290026)

[5 On MPEG/JPEG file embedding (MPEG#141, Issue#87) 12](#_Toc158290027)

[5.1 Discussion 12](#_Toc158290028)

[5.2 Initial text proposal 12](#_Toc158290029)

[6 Extending pixi for more use-cases (MPEG#143-144, Issue#95 and #109) 12](#_Toc158290030)

[6.1 Pixel information 13](#_Toc158290031)

[6.1.1 Definition 13](#_Toc158290032)

[6.1.2 Syntax 14](#_Toc158290033)

[6.1.3 Semantics 14](#_Toc158290034)

[7 Disparity adjustment property for frame-packed stereo pair (MPEG #144, issue #111) 15](#_Toc158290035)

[8 Stereo aggressors item property (MPEG #145, issue #121) 15](#_Toc158290036)

[8.1 Abstract 15](#_Toc158290037)

[8.2 Introduction 16](#_Toc158290038)

[8.3 Proposal 16](#_Toc158290039)

[8.3.1 Stereo aggressors item property 16](#_Toc158290040)

[9 Adding monoscopic signaling to stereo pair group (MPEG #145, issue #122) 18](#_Toc158290041)

[9.1 Abstract 18](#_Toc158290042)

[9.2 Introduction 18](#_Toc158290043)

[9.3 Proposal 18](#_Toc158290044)

[9.4 Alternative 19](#_Toc158290045)

# 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

A black background with white squares

Description automatically generated

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#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 (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) and MPEG #145, issue [#128](https://git.mpeg.expert/MPEG/Systems/FileFormat/HEIF/-/issues/128))

*[[ Ed. (LB): MPEG#145: How do we unify the two approaches? (Entity group and derived region item)]]*

## Discussion

### *Motivation, use cases and initial proposal*

The HEIF 2nd edition specification [1] enables associating an annotation with a region of an image by defining a region item associated with an image item. Several regions of an image can be described either within a same region item (typically when the regions share the same annotations) or within different region items (typically when the regions do not share the same annotations).

Current specification does not provide specific tools for combining multiple annotated regions and annotating the corresponding union as a whole, or for signaling the aggregation of several regions forming logical parts of a same ‘object’. When an image is annotated, such tools are useful to document in the file format explicit relationships between regions rather than possibly inferring them at player-side.

The Figure 1 illustrates the former use case, when there are several regions inside an image, for instance, each corresponding to a different person, it is possible to associate the name of each person with the corresponding region by defining multiple region items (illustrated by identifiers ranging from 1 to 7) and their respective ‘label’ annotations, for instance using a UserDescriptionProperty. However, it is not possible to easily combine those annotated regions corresponding to different persons and to associate an annotation applying to the union as a whole, for instance indicating that they belong to the same family and possibly providing overall information on the family.

A black and white drawing of a group of people

Description automatically generated

Figure 1: Example of a region defined as a union of annotated regions

To address this former use case, a new type of derived region item representing the union of several regions defined as inputs is proposed.

The Figure 2 illustrates the second use case where a content creator wants to indicate that an ‘object’ in an image represented by a region is also a logical aggregation of several other regions of the image. The aggregation does not imply that the area of the region corresponding to the ‘object’ is the exact combination of areas of other aggregated regions. In Figure 2, a first region (in green) corresponds to a person and several other regions correspond to the head, body, arms and legs of this person respectively. However, it is not currently possible to indicate the relationship between the head, body, arm and leg regions and the region representing the whole body.

A screenshot of a computer

Description automatically generated

Figure 2: Example of relation between regions

To address this second use case, a new entity to group is proposed to group region items and to indicate that the ‘object’ covered by the region defined by the first entity of the group is logically including the areas covered by the regions described by the other entities of the group.

### *Responses to open questions*

First, we recall the semantics implied by region item and masks. According to HEIF specification, the region annotation associated with a region item applies to each region described in the region item individually. If the region is described by a mask (either embedded in the region item or in a separate mask item), the mask indicates which pixels are part of the region, meaning that a mask describes one single region, possibly comprising disjoint areas.

#### Open Question 1

At MPEG#141, the following question has been raised:

* “*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?*”

Regarding this question, indeed a mask could represent a region composed of disjoint areas and thus it could be used for representing a region composed of disjoint areas where each area corresponds to a member of the same family. Annotations associated with the region item using that mask then apply to the region described by the mask as a whole and actually would document family information, but additional region items representing each person are still needed to annotate each person individually.

Such approach has following drawbacks compared to the proposed derived region item:

* A mask approach does not allow creating an explicit and direct link between the region item (mask) representing the family and each region item representing each person of the family individually. This relationship may be inferred, e.g., by comparing the pixel areas covered by the masks and each individual region items, but this is more complex for the player than providing an explicit signalling of the relationship.
* A mask approach implies to encode a mask to represent the union of regions representing the family. This is more costly in terms of required data and more complex than using item references to document the union of region items composing the family.

A second alternative was also suggested during MPEG#141 consisting of defining a region item representing the family and comprising a list of regions, one region for each member of the family. As above alternative, this would allow associating annotations (e.g., surname) dedicated to the family with each region individually. But it would still require region items dedicated to each person to associate them with annotations dedicated to the person (e.g., first name). Again, such approach does not allow creating an explicit link between the region item describing the family as a whole and each region item describing each person of the family. This second alternative has same drawbacks as above alternative (including duplicate geometry information needed to describe each region individually and regions in the union).

A third alternative could be to associate both annotations dedicated to the family and annotations dedicated to a person with each region item representing a person.

But this third approach does not allow identifying easily all members of a family. Indeed, this would require to infer the union representing the family by parsing each entry of the ItemPropertyAssociationBox to check whether the corresponding region item is associated with a same annotation identifying the family.

Therefore, we think that the approach based on derived region item represents the best approach to represent a union of regions and to annotate this union as a whole.

#### Open Question 2

At MPEG#141, the following question has been raised:

* “*Is there a restriction for any of the proposals that the separate regions must all be derived from a single image item?*”

It can be noted that region items associated with images representing different visual contents do not share a common referential. Only region items associated with images representing the same visual content have a common referential.

We think that the separate regions referred by the derived region item or the entity group must all be derived from region items sharing a common reference space, i.e., at least from images having a same visual content.

## Proposal

For convenience, we provide below the proposed specification text from TuC with some editorial improvements.

### Proposal 1: Union of regions

Add the following section in section 11.3.3.2 *Derived region item types*

## 11.3.3.2.2 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 11.3.2 and 11.2.1.

### Proposal 2: Relations between region items

Add the following section in section 11.3 *Regions and region annotations for an image item*

## 11.3.4 Region Entity Group

## 11.3.4.1 ‘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 body with regions corresponding to body parts (e.g., the head, legs or arms of the body) to indicate that the body is logically including the body parts.

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.



# 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 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.

# Extending pixi for more use-cases (MPEG#143-144, [Issue#95](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/95) and [#109](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/109))

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). It was further updated in [m64755](https://dms.mpeg.expert/doc_end_user/current_document.php?id=89376). During MPEG#144 it was discussed that adding chroma siting information might also be beneficial and that this could be done by extending the existing table in CICP with more subsamplings. The proposal from m64755 is included below, with the changes mentioned in issue [#109](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/109) included.

The tables for subsampling\_type and subsampling\_location should ideally be added to the CICP spec and instead referenced from HEIF.

## Pixel information

### Definition

|  |  |
| --- | --- |
| Box type: | 'pixi' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per item): | No, unless 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/depth 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 1 is used:

0x000001 has\_alpha: indicates that the coded item contains both color/monochrome pixels *and* alpha.

0x000002 alpha\_is\_premultiplied: 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 has\_alpha is non-zero. If this flag is used, has\_alpha shall be set.

0x000004 has\_subsampling: indicates that the PixelInformationProperty contains subsampling information.

### Syntax

aligned(8) class PixelInformationProperty  
extends ItemFullProperty('pixi', version, px\_flags) {  
{  
 if (version == 0) {  
 unsigned int(8) num\_channels;  
 for (i=0; i<num\_channels; i++) {  
 unsigned int(8) bits\_per\_channel;  
 }  
 }  
 else if (version == 1) {  
 unsigned int has\_alpha = (px\_flags & 1);  
 unsigned int alpha\_is\_premultiplied = (px\_flags & 2);  
 unsigned int has\_subsampling = (px\_flags & 4);  
 unsigned int(8) num\_channels;  
 for (i=0; i<num\_channels; i++) {  
 unsigned int(3) channel\_idc;  
 unsigned int(2) channel\_data\_type;  
 unsigned int(1) channel\_label\_present;  
 unsigned int(2) reserved;  
 unsigned int(8) bits\_per\_channel;  
 if (has\_subsampling) {  
 unsigned int(4) subsampling\_type;  
 unsigned int(4) subsampling\_location;  
 }  
 if (channel\_label\_present) {  
 utf8string channel\_label;  
 }  
 }  
 }  
}

### 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 and interpretation for this field depend on the value of the channel\_data\_type.

has\_alpha: If set to 1, one of the channels shall have a channel\_idc set to 1.

alpha\_is\_premultiplied:

has\_subsampling:

channel\_idc: This field indicates the contents of the channel. A value of 0 indicates colour/grayscale. A value of 1 indicates alpha. A value of 2 indicates depth. Values 3-8 are reserved for future use. At most one channel shall have a channel\_idc of 1.

channel\_data\_type: This field indicates the data type of the channel. A value of 0 indicates unsigned integers. A value of 1 indicates signed integers. A value of 2 indicates IEEE 754 floating-point. The value 3 is reserved. When set to 2, bits\_per\_channel shall take a value of 16,32, 64 or 128.

channel\_label\_present: If set to 1, channel\_label is present for this channel.

subsampling\_type: This field indicates how the channel is subsampled relative to the unsubsampled channels. If has\_subsampling is not set, a default value of 15 shall be assumed. If has\_subsampling is set, at least one channel shall have a subsampling\_type of 0. A value of 0 indicates no subsampling. A value of 1 indicates 2x horizontal subsampling (4:2:2). A value of 2 indicates 2x horizontal and vertical subsampling (4:2:0). A value of 3 indicates 4x horizontal subsampling (4:1:1). A value of 4 indicates 2x vertical subsampling (4:4:0). Values 5-14 are reserved. A value of 15 indicates that subsampling for this channel is undefined.

subsampling\_location: This field indicates the location of the channel samples relative to the unsubsampled channels. If subsampling\_type is 0, this field shall have a value of 2. If subsampling\_type is 15, this field shall have a value of 15. Values 6-14 are reserved. Values are converted to a horizontal and vertical offset compared to unsubsampled channels given the table below, where type is the value of subsampling\_type.

|  |  |  |  |
| --- | --- | --- | --- |
| value | type == [1,3] (4:2:2 & 4:1:1) | type == 2 (4:2:0) | type == 4 (4:4:0) |
| 0 | 0.0, 0.0 | 0.0, 0.5 | 0.0, 0.5 |
| 1 | 0.5, 0.0 | 0.5, 0.5 | 0.0, 0.5 |
| 2 | 0.0, 0.0 | 0.0, 0.0 | 0.0, 0.0 |
| 3 | 0.5, 0.0 | 0.5, 0.0 | 0.0, 0.0 |
| 4 | 0.0, 0.0 | 0.0, 1.0 | 0.0, 1.0 |
| 5 | 0.5, 0.0 | 0.5, 1.0 | 0.0, 1.0 |
| 6-14 | Reserved | Reserved | Reserved |
| 15 | Undefined | Undefined | Undefined |

channel\_label: This field is a NULL-terminated string that provides a description of the channel contents.







# Disparity adjustment property for frame-packed stereo pair (MPEG #144, issue [#111](https://mpeg.expert/software/MPEG/Systems/FileFormat/HEIF/-/issues/111))

During MPEG #144, a new item property that describes disparity adjustment for a stereo pair entity group was proposed. This property was accepted into the (new) working draft for 3rd edition amendment 2. During the discussion, it was mentioned that this property would also be useful for a frame-packed stereo pair as described by OMAF and that the concept of frame-packed stereo pair items should potentially be moved from OMAF to HEIF.

One objection was raised on this that the current StereoVideoBox in OMAF is overly complicated and a bit wasteful and that rather than simply migrate it from OMAF it might be better to define a new dedicated box.

# Stereo aggressors item property (MPEG #145, issue [#121](https://git.mpeg.expert/MPEG/Systems/FileFormat/HEIF/-/issues/121))

*[Ed.(LB): A question was raised on what a player should do when it encounters a stereo aggressor. Since they are very subjective and may depend a lot on the device and person, it's very hard to add normative language. We could however consider adding a generic recommendation like "if a stereo aggressor is present and has a severity of 0 (unspecified) or >= 64, the stereo pair should not be displayed as stereo in a stereoscopic environment unless the player has a method for mitigating the aggressor"]*

## Abstract

The conditions when capturing a stereo image sometimes lead to stereoscopic viewing issues known as stereo aggressors. These issues may cause discomfort when the stereo pair is viewed on a stereoscopic display. There currently is no HEIF metadata available to signal this to parsers. This contribution proposes to add a new item property that allows signaling of various types of stereo aggressors.

## Introduction

There are multiple types of issues during stereo image capture that can cause viewer discomfort when the image is viewed on a stereoscopic display. These issues are commonly referred to as stereo aggressors.

These aggressors can have different causes. For example:

* Objects that are too close to the cameras
* Occlusions on one of the cameras
* Varying lighting conditions or lens characteristics on one of the cameras

There currently is no way to signal if such issues were detected at capture time.

In many cases it's enough to know if an aggressor was encountered or not for a stereo pair, but it may also be useful to be able to pinpoint more exactly where in the images the issue was detected.

This proposal ties into proposal m65965 [1] which allows a file writer to specify which image should be displayed in a monoscopic viewing environment.

## Proposal

We propose to add a new item property that can be used to signal the presence of stereo aggressors.

The item property is meant to be associated with a stereo pair group and indicates whether an aggressor has been detected for that pair or not.

In addition to the stereo pair group, the item property may also be associated with one or both of the images in the group to indicate which image the issue is present in. For example, if a lens occlusion is detected in the left image, a stereo aggressor property can be associated both with the stereo pair group and the left image item. If an even more fine-grained location is desired, the item property may be associated with a region item associated with one or both of the image items in the pair.

NOTE: An image may belong to multiple stereo pairs. Even though an image item may be associated with a stereo aggressor property, that property may not apply to all the stereo pairs that the image item belongs to. This is the reason why all stereo aggressor properties need to also be associated with a stereo pair group.

### Stereo aggressors item property

#### Definition

|  |  |
| --- | --- |
| Box type: | 'stag' |
| Property type: | Descriptive item property |
| Container: | ItemPropertyContainerBox |
| Mandatory (per associated item): | No |
| Quantity (per associated item): | 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 shall be used with a stereo pair. It may 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, 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: 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.

#### 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 is a hint that indicates how severe the aggressor is considered to be. A value of 0 indicates an unspecified or unknown severity. A value of 1 indicates a very mild severity. A value of 127 indicates an extreme severity.

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. If not present, it defaults to the empty string.

# Adding monoscopic signaling to stereo pair group (MPEG #145, issue [#122](https://git.mpeg.expert/MPEG/Systems/FileFormat/HEIF/-/issues/122))

## Abstract

The current specification for the 'ster' entity group does not sufficiently signal how to display the file on a monoscopic display. In this contribution, we propose an enhancement to allow file creators to specify either one image from the pair, or a separate optimized image for monoscopic display.

## Introduction

Stereoscopic displays are currently not very common, so any stereo images captured need to consider how they should be displayed on monoscopic displays. The current language for the 'ster' entity grouping does not provide adequate signaling for a file creator to indicate how the pair should be displayed on a monoscopic display. The only way of indicating which item in the stereo pair should be used on a monoscopic display is by making it the primary item. But this does not work when a file contains multiple stereo pairs.

At a minimum, it should be possible for a file creator to indicate whether the left *or* the right image should be displayed on a monoscopic display. However, this basic capability alone may not suffice for more complex configurations.

A stereo capture device may be equipped with a left, right *and* center camera. Such a device may want the left and right images to be used when displaying on a stereoscopic display, but the center image to be displayed on a monoscopic display.

Another use-case is where either the images in the stereo pair or the monoscopic image is the result of fusion of multiple cameras or heavy processing. When this is the case, you may want the monoscopic image to be a less-processed version of one of the stereo pair images or some kind of fusion of them.

We propose to add a version = 1 of the 'ster' EntityToGroupBox that allows for three input images (left, right and mono), as well as some flags to indicate how the mono image relates to the stereo pair.

## Proposal

We propose to change clause 6.8.5 to the following:

The stereo pair entity grouping ('ster') indicates a set of images that form a stereo pair suitable for displaying on a stereoscopic display. The entity group shall contain entity\_id values that point to image items and shall contain no entity\_id values that point to tracks. The first entity\_id value (with i equal to 0) indicates the left view and the second entity\_id value indicates the right view.

If version is 0, flags shall be 0 and the entity group shall contain exactly two entity\_id values that point to image items.

If version is 1, the following holds:

* + The entity group shall contain exactly three entity\_id values that point to image items.
  + The third entity\_id value (with i equal to 2) indicates the image to display on a monoscopic display. This may be the same as either the first or second entity\_id.
  + The value of (flags & 3) indicates how the third entity\_id relates to the stereo pair. The following values are defined:  
     0: Unknown  
     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 'ster' version 1, file creators may add both a 'ster' version 1 and a 'ster' version 0 to the file. An 'altr' group may then be added containing the version 1 and 0 group id's in that order to indicate that only one should be used. In order to use group id's in an 'altr' group, the 'unif' brand is required.

## Alternative

One alternative considered was something along these lines:

Image item 1: center image  
Image item 2: left image  
Image item 3: right image  
'ster' group 4: [2, 3]  
'altr' group 5: [4, 1]

One way of parsing this is "*select either the stereo pair or the center image*". As noted in GitHub [issue #84](https://github.com/MPEGGroup/FileFormat/issues/84), it is however very unclear how 'altr' groups containing a mix of groups and images should be interpreted.

Even if the interpretation above is correct, it causes practical problems when implemented. Most existing image processing libraries and APIs deal with image indexes, not groups. If queried what the image count is in the example above, 1 (for mono display), 2 (for stereo display) and 3 (all images in the file) are all valid answers. If asked to decode the first image in the file, it's very unclear what that means.