**A black and white logo

Description automatically generated** **ISO/IEC JTC 1/SC 29/WG 03 N1243**

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

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title: Technology under consideration on MIAF**

**Status:** Approved

**Date of document:** 2024-04-26

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

**Expected action:** ACT

**Action due date:**

**No. of pages:** 4 (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 N1243**

**Rennes, FR – April 2024**

|  |  |
| --- | --- |
| **Title** | **Technology under consideration on MIAF** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **23853** |

# Introduction

This document captures the technologies under consideration for future amendments of the Multi-Image Application Format (MIAF), ISO/IEC 23000-22 specification.

The current technologies under consideration include:

* Add informative annex documenting decisions made in implementations.
* Add recommendation on 'tmap' derived item input orientation.

# Add informative annex documenting decisions made in implementations

# Abstract

Some features in HEIF are difficult to surface in image decoding libraries that deal with multiple types of images. Those libraries will typically provide a simplified view of what is in a file to clients. The decisions that go into creating that simplified view is currently not documented anywhere, even though many implementations have converged on the same decisions. Here MIAF adds an informative annex to document these decisions.

# 2.2 Introduction

We have identified two cases that could benefit from documenting what existing implementations do.

1. HEIF sequence loop counter
2. Image item and 'altr' group sort order

# 2.3 HEIF sequence loop counter

## 2.3.1 Introduction

Animated image formats typically have some way of specifying how repeats should be handled. This is typically done via metadata specifying a loop counter. The values that the loop counter can take is typically:

|  |  |
| --- | --- |
| Loop counter value | Meaning |
| None | Unspecified, behaviour depends on implementation |
| 0 | Play indefinitely |
| 1 | Play once |
| N | Play N times |

When the loop counter value is none, most implementations have chosen this to mean loop indefinitely since this is what is done for older animated image formats.

## 2.3.2 Loop counter in HEIF

Looping in HEIF sequences is controlled via the EditListBox, in particular via the RepeatEdits flag (ISOBMFF 8.6.6.1):

When the TrackHeaderBox duration is not indefinite (all 1s), then the edit list is repeated R times such that the total duration of the edit list multiplied by R equals the TrackHeaderBox duration (R is not necessarily an integer). If the TrackHeaderBox duration is indefinite, then the edit list is repeated indefinitely.

All browsers that support some flavor of HEIF sequences have converged on the following mapping from HEIF to the table above:

|  |  |
| --- | --- |
| Case | Loop counter value |
| No EditListBox | None |
| EditListBox present, RepeatEdits is 0 | 1 |
| RepeatEdits is 1, TrackHeaderBox duration is all 1 | 0 |
| RepeatEdits is 1, TrackHedearBox duration is not all 1 | ceil(R) |

# 2.4 Image item and 'altr' group sort order

## 2.4.1 Introduction

(See MIAF issue [#37](https://git.mpeg.expert/MPEG/Systems/ApplicationFormat/MIAF/-/issues/37))

Neither HEIF nor MIAF tells a parser how a file with multiple images should be represented. This is good, since particular environments may have specific things in mind with files. In some cases, they may only want to surface the primary item, in others they may want to sort according to some specific metadata like a timestamp or similar.

## 2.4.2 Sort order without 'altr' groups

In the absence of an externally mandated sort order, most HEIF parsers tend to represent a file with multiple displayable main image items as an array of images sorted by item ID. In other words, for a file with image items with IDs [1000,2000,3000,4000], a parser would say:

* This is a file with 4 images.
  + image index 0 => item 1000
  + image index 1 => item 2000
  + image index 2 => item 3000
  + image index 3 => item 4000

## 2.4.3 Sort order with 'altr' groups

It is not at all clear what "default" sort order to use once we have 'altr' groups in a file.

Assume we have a file with the following structure:

* image item 10
* image item 20
* image item 30
* image item 40
* image item 50
* altr group 15 containing [50,10]
* altr group 25 containing [20,40]

A parser that "handles" the altr groups should report that the file contains three images. It may also report that two of those images have alternative representations available. But it's unclear how the image items and groups should be sorted. Some alternatives:

1. altr groups are sorted according to the lowest item ID in the group

* index 0 => altr group 15
* index 1 => altr group 25
* index 2 => image item 30

1. altr groups are sorted according to the highest item ID in the group (but still from low to high in sort order)

* index 0 => image item 30
* index 1 => altr group 25
* index 2 => altr group 15

1. altr groups are sorted according to the item ID of the first item in the group

* index 0 => altr group 25
* index 1 => image item 30
* index 2 => altr group 15

1. altr groups are sorted according to the ID of the group itself

* index 0 => altr group 15
* index 1 => altr group 25
* index 2 => image item 30

The behaviour we feel is most logical is option 4. While the group ID affecting sort order may feel a bit unexpected, it makes for a very simple rule for both readers and writers: Everything is sorted according to ID.

The other nice thing about option 4 is that it's unaffected by what features the parser supports. If items 20 and 50 are derived items not supported by the parser, it may not surface these items at all. If the sort order is handled at a higher level, that level may not get told that these items exist, in which case it can't correctly implement options 1 to 3.

# 2.5 Proposal

The following new informative annex is being added to MIAF.

# Annex B

(informative)

# Implementations

## B.1 Introduction

This annex documents decisions made in implementations.

## B.2 HEIF sequence looping

Looping in HEIF sequences is specified via the EditListBox, in particular via the RepeatEdits flag (ISOBMFF 8.6.6.1):

When the TrackHeaderBox duration is not indefinite (all 1s), then the edit list is repeated R times such that the total duration of the edit list multiplied by R equals the TrackHeaderBox duration (R is not necessarily an integer). If the TrackHeaderBox duration is indefinite, then the edit list is repeated indefinitely.

All browsers that support some flavor of HEIF sequences have converged on the following behaviour:

|  |  |
| --- | --- |
| Case | Behaviour |
| No EditListBox | Play indefinitely |
| EditListBox present, RepeatEdits is 0 | Play once |
| RepeatEdits is 1, TrackHeaderBox duration is all 1 | Play indefinitely |
| RepeatEdits is 1, TrackHedearBox duration is not all 1 | Play ceil(R) times |

## B.3 HEIF image item sort order

For a HEIF image collection containing multiple displayable main images, most implementations will surface this as a list of images. In the absence of an externally mandated sort order, most implementations will sort by increasing item ID.

In other words, for a file with image items with IDs [1000,2000,3000,4000], it would be surfaced as follows:

|  |  |
| --- | --- |
| Image index | Item ID |
| 0 | 1000 |
| 1 | 2000 |
| 2 | 3000 |
| 3 | 4000 |

## B.4 HEIF 'altr' group sort order

For files that contain 'altr' groups, the following rule is sometimes used for the default sort order:

All image items belonging to an 'altr' group are replaced by that group. Items **and** groups are sorted by increasing ID.

In other words for a file containing the following:

 image item with ID 10

 image item with ID 20

 image item with ID 30

 image item with ID 40

 image item with ID 50

 altr group 15 containing image items [50,10]

 altr group 25 containing image items [20,40]

It would be surfaced as follows:

|  |  |
| --- | --- |
| Image index | Contents |
| 0 | 'altr' group 15 |
| 1 | 'altr' group 25 |
| 2 | Image item 30 |

# Add recommendation on 'tmap' derived item input orientation

# 3.1 Abstract

The tone-map derived item in the HEIF 3ed Amd2 preliminary working draft does not have any restrictions on the orientations of the input images. Here we add a recommendation to MIAF that states that both inputs should have the same 'imir' and 'irot' properties as well as be contained within an 'altr' group for backwards compatibility.

# 3.2 Introduction

The tone-map derived item takes a base image (the first input image) and applies a gain map (the second input image) to it to go from an SDR or HDR image to a target HDR or SDR image. If the base and gain map inputs have the same orientation, this reconstruction can be done without applying the orientation transformation, which can then potentially be handled by the GPU once the reconstructed image is displayed.

HEIF does not restrict the orientation of the input images. Here we add this as a recommendation to MIAF as that will lead to files that are easier to deal with and can be displayed in a more performant manner.

We can also add a restriction to MIAF that states that 'tmap' derived items shall be contained within an 'altr' group for backwards compatibility.

# 3.3 Proposal

Add the following subclause to MIAF:

#### 7.3.11.5 Tone-map derived image

A MIAF file may include tone-map derived images, i.e. derived image items with item\_type value 'tmap'.

All input images of a tone-map image item should have the same rotation and mirroring transformative properties.

If a tone-map derived item is present in a MIAF file, it shall be contained in an alternative images group together with another image item that is a valid MIAF master image item.