INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2015/M35844
February 2015, Geneva, Switzerland

Source	**Telecom ParisTech, Canon Research Centre France**
Status	**For consideration at the 111th MPEG Meeting**
Title	**Review of use cases on the Image File Format**
Author	Cyril Concolato, Jean Le Feuvre, Franck Denoual, Frédéric Mazé, Eric Nassor

1 Introduction

This contribution compares the use of the Image File Format for some particular use cases using the tools from the output of the 109th meeting and from the output the 110th meeting (study text). Between these two versions, various image descriptors were classified into descriptive versus prescriptive metadata (discussions during MPEG 110th). We propose a merge of existing tools that covers these use cases and complies with the descriptor classification (descriptive vs. prescriptive).

2 Defect Consideration

In the output of Sapporo, the 'simr' item references are from the 'simd' metadata to the image item it impacts. Although it allows a better compaction of the references, the reference should be inverted to follow the common practice (from the item to the one that describes it) so that, for an image, it is easier to access its descriptive metadata. This was not noted in the FRNB comments. This contribution assumes that item reference direction for simd are in the usual way, from the item to the 'simd'.

Replace
"The image item is associated to an ISOBMFFMetaData item through an item reference of type 'simr' from the ISOBMFFMetaData item to the image item. It is allowed to have several items sharing the same ISOBMFFMetaData item.
It is allowed for an image item to have multiple 'simr' references from different ISOBMFFMetaData items.
"

with

"The image item is associated to an ISOBMFFMetaData item through an item reference of type 'simr' from the image item to the ISOBMFFMetaData item. It is allowed to have several items sharing the same ISOBMFFMetaData item.
It is allowed for an image item to have multiple 'simr' references from different ISOBMFFMetaData items. If the same child box of ISOBMFFMetaData appear several time in the ISOBMFFMetaData referenced, only the last one found, in the indicated order of item references, shall be used.

”

3 Use cases and comparisons

In this section we will review the storage cost between the different versions, in terms of number of items and number of item references. Keep in mind that increasing the number of items imply increasing the size of at least ‘iinf’ and ‘iloc’ tables.

Sapporo output	Strasbourg output

3.1 Simple Image transformation

We address here the description of an image item and its clap, rotated version (with multiples of 90° rotation).
With output of 109th meeting, one cannot describe rotation operation. We then propose a modification of the simd descriptor to support rotation operation and then compare with study text.

The proposed inclusion of the rotation operation in the image metadata descriptor ‘simd’ is as follows:

```
aligned(8) class ISOBMFFMetaData {
   CleanApertureBox  clap;              // optional
   PixelAspectRatioBox pasp;            // optional
   ColourInformationBox colour;         // optional
   ImageSpatialRelationBox location;    // optional
   ImageRotationBox rotation;           // optional
   Box extra_boxes[];                   // optional
}
```

```
aligned(8) class ImageRotationBox
extends FullBox('irot', version = 0, flags = 0) { // 12 extra-bytes
   unsigned int (6) reserved = 0;
   unsigned int (2) angle;
}
```

We can see from the comparison below that descriptions are possible with both approaches.

Output of 109th with rotation support	Study text

item_type = ‘hvc1’, itemID=1 // the image
item_type = ‘hvcC’, itemID=2 // config data
item_type = ‘simd’, itemID=3 // operation descr.,
 	
Item Location:
itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
 	
Item Reference:
type=’init’, fromID=1, toID=2;
type=’simr’, fromID=1, toID=3;
	item_type = ‘hvc1’, itemID=1 // the image
	item_type = ‘hvcC’, itemID=2 // config data
	item_type = ‘irot’, itemID=3 // operation descr.,
	item_type = ‘clap’, itemID=4 // operation descr.,
	
	Item Location:
	itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
	itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
	itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
	itemID = 4, extent_count = 1, extent_offset = T, extent_length = U;

Media data box:
HEVC Image (at file offset X, with length Y)
HEVC Config Record (at file offset P, with length Q)
Image descriptor data block (at file offset R, with length S)
	Item Reference:
	type=’init’, fromID=1, toID=2;
	type= ’dimg’, fromID=3, toID=4; //irot
	type= ’dimg’, fromID=4, toID=1; //clap
	
	Media data box:
	HEVC Image (at file offset X, with length Y)
	HEVC Config Record (at file offset P, with length Q)
	Clap value (at file offset T, with length U)
	Rotation value (at file offset R, with length S)

3 items / 2 item references: • an 'hvcC' item • an 'hvc1' item for the initial non-rotated image, with an 'init' reference to 'hvcC' • a ISOBMFFMetaData 'simd' item with a 'simr' item reference to the 'hvc1' item Primary item: only the 'hvc1' item Problem: the rotated image cannot be listed as a primary item.	4 items / 3 item references: • an 'hvcC' item • an 'hvc1' item for the initial non-rotated image, with an 'init' reference to 'hvcC' • a derived item 'irot' with a 'dimg' item reference to the 'clap' item • a derived item 'clap' with a 'dimg' item reference to the 'hvc1' item Primary item: either the 'hvc1' or 'irot' or ‘clap’ item

Although the box’ed version is slightly bigger than the ‘irot’ item (12 bytes for full box header), the benefit of using this approach is clear when combining transformations, such as rotation and CleanApperture, since only one ‘simd’ is needed, rather than a cascade of derived items. If more effects were to be defined for an image, more items would have to be declared with the study text, whereas our approach would only require modifying the ISOBMFFMetaBox.

As explained in FRNB, we suggest keeping a generic derived item, ‘dimg’, referencing both the image item and the metadata describing effects that shall be applied to it. Such an item could then be listed as a primary item.

Another benefit of the approach is that an author can clearly indicate that it only wants the rotated item to be displayed, by using ‘hvc1’ item and ‘simd’ metadata, or have a choice between both by using ‘hvc1’ item for the non-rotated image and ‘dimg’ + ‘simd’ items for the rotated one. The study text approach is ambiguous, as both the ‘hvc1’ and ‘irot’ items could be used.

We therefore suggest removing ‘irot’ ad ‘clap’ items and introduce a ‘irot’ box in the ISOBMFFMetaData.

3.2 N Images with the same rotation

N+2 items / 2xN item references (in 2xN boxes): • an 'hvcC' item • N 'hvc1' items, each with an 'init' reference to the same 'hvcC', and a	2xN+1 items / 2xN item references (in 2xN boxes): • an 'hvcC' item • N 'hvc1' items each with an 'init' reference to the same 'hvcC'

'simr' reference to the 'simd' item • a single 'simd' item Primary item: one of the 'hvc1' items Problem: the number of item reference boxes is large.	• N derived items 'irot', each with a 'dimg' item reference to the corresponding 'hvc1' item Primary item: one of the 'hvc1' or 'irot' items Problem: the number of item reference boxes is large. Problem: The number of items and of item references is the double of the number of images.

Note that the same problem applies when 'clap' information is reused between images.

3.3 Image composition

One image composed of N other images.

Output of 109th	Study text

Example composing 2 images
item_type = ‘hvc1’, itemID=1 // 1st image
item_type = ‘hvc1’, itemID=2 // 2nd image
item_type = ‘hvcC’, itemID=3 // config data
item_type = ‘simd’, itemID=4 // compo info for 1st img
item_type = ‘simd’, itemID=5 // compo info for 2nd img
Item Reference
type=’init’, fromID=1, toID=3;
type=’init’, fromID=2, toID=3;
type=’simr’, fromID=1, toID=4;
type=’simr’, fromID=2, toID=5;
Item Location
itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
itemID = 4, extent_count = 1, extent_offset = T, extent_length = U;
itemID = 5, extent_count = 1, extent_offset = V, extent_length = W;
	Example composing 2 images
	item_type = ‘hvc1’, itemID=1 // 1st image
	item_type = ‘hvc1’, itemID=2 // 2nd image
	item_type = ‘hvcC’, itemID=3 // config data
	item_type = ‘iovl’, itemID=4 // compo info
	
	Item Reference
	type=’init’, fromID=1, toID=3;
	type=’init’, fromID=2, toID=3;
	type=’dimg’, fromID=4, toID=1,2;
	
	Item Location
	itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
	itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
	itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
	itemID = 4, extent_count = 1, extent_offset = T, extent_length = U;

2xN+1 items / 2xN item references (in 2xN): • an 'hvcC' item • N 'hvc1' items, each with an 'init' reference to the 'hvcC' and a 'simr' reference to the 'simd' item • N 'simd' items	N+2 items / N+1 item references (in N+1 boxes): • an 'hvcC' item • N 'hvc1' items each with an 'init' reference to the 'hvcC' • 1 derived item 'iovl', with a 'dimg' item reference to all 'hvc1' items

Primary item: one of the 'hvc1' items Problem: The number of items and of item references is the double of the number of images; no access to composed image. Problem: no access to composed image (but solved if using ‘dimg’ item as proposed’)	Primary item: one of the 'hvc1' items or the 'iovl' item Problem: positioning and z-ordering is derived from the iref box, not robust to re-writing of the file (removing of an item requires rewriting the iref AND the iovl).

As can be seen, the ‘iovl’ introduced in the study text is quite efficient for simple image composition (less items). However it makes the editing (removal or insertion of one item) more difficult since the ‘iovl’ item would have to be re-written. Using in approach with ‘simd’ is closer to the design of the file format where editing of a track layout does not impact other tracks’ layouts.

3.4 Tiling description

Let’s consider the description of a simple image with 2 tiles. Image and tile items are described as image item information, but tiling description differs. This description can be useful for fast access to a region of interest.

Output of 109th	Study text (assuming all tiles have same size)

item_type = ‘hvc1’, itemID=1 // the image
item_type = ‘hvcC’, itemID=2 // config data
item_type = ’hvt1’, itemID=3 // Tile 1
item_type = ’hvt1’, itemID=4 // Tile 2
item_type = ‘simd’, itemID=5 // Tile 1 descriptor
item_type = ‘simd’, itemID=6 // Tile 2 descriptor
 	
Item Reference:
type=’init’, fromID=1, toID=2;
type=’tbas’, fromID=3, toID=1;
type=’tbas’, fromID=4, toID=1;
type=’simr’, fromID=3, toID=5;
type=’simr’, fromID=4, toID=6;
 	
Item Location:
itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
itemID = 4, extent_count = 1, extent_offset = T, extent_length = U;
itemID = 5, extent_count = 1, extent_offset = V, extent_length = W;
itemID = 6, extent_count = 1, extent_offset = A, extent_length = B;
 	
Media data box:
HEVC Image (at file offset X, with length Y, contains tiles data)
HEVC Config Record (at file offset P, with length Q)
	item_type = ‘hvc1’, itemID=1 // the image
	item_type = ‘hvcC’, itemID=2 // config data
	item_type = ’hvt1’, itemID=3 // Tile 1
	item_type = ’hvt1’, itemID=4 // Tile 2
	item_type = ‘iovl’, itemID=5 // tiling positions
	item_type = ‘simd’, itemID=6 // Tiles size
	
	Item Reference:
	type=’init’, fromID=1, toID=2;
	type=’tbas’, fromID=3, toID=1;
	type=’tbas’, fromID=4, toID=1;
	type=’dimg’, fromID=5, toID=3, 4; // tile compo
	type=’simr’, fromID=3, toID=6; // tiles size
	type=’simr’, fromID=4, toID=6; // tiles size
	
	Item Location:
	itemID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
	itemID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
	itemID = 3, extent_count = 1, extent_offset = R, extent_length = S;
	itemID = 4, extent_count = 1, extent_offset = T, extent_length = U;
	itemID = 5, extent_count = 1, extent_offset = V, extent_length = W;
	itemID = 6, extent_count = 1, extent_offset = A, extent_length = B;
	
	Media data box:
	HEVC Image (at file offset X, with length Y, contains tiles data)

2 Image descriptors (at file offsets V, A, with resp. length W, B)	HEVC Config Record (at file offset P, with length Q) 1 Overlay descriptor (at file offset V, with length W) 1 Image descriptor (at file offset A, with length B)
Let T be the number of tiles and N the number of images: N*(2*T +1) + 1 item declarations N*(2*T+1) item references Primary item: one of the 'hvc1' or ‘hvt1’ items	Let T be the number of tiles and N the number of images (best case: all tiles with same size): N*(T+3) + 1 item declarations* N*(2*T+1) + 1 item references (*) iovl size is a function of T (offsets) Primary item: the 'hvc1' or ‘hvt1’ items (or the 'iovl' item which is the same as hvc1) Worst case: each tile with different size: N*(2*T+2) item declarations N*(2*T+2) item references

As can be seen, ‘iovl’ could be used for tiling composition but not for tile description. For an hvt1 item, only its sizes can be retrieved through the simr reference. To obtain the tile position, it requires parsing the overlay, the list of images referencing this ‘iovl’ and then get the x, y position for the tile, x, y here corresponding to a position in a composite picture. These positions may differ from the original x, y positions of the tiles. As such, we propose to revert to tile positions description in ‘simd’ since these are descriptive metadata for a tile. By doing so, we clearly separate tile description (in ‘simd’) from tile composition (in ‘iovl’).

We therefore recommend using the Sapporo output ‘isre’ rather than the study text version ‘ispe’

4 Additional Proposal

It seems quite annoying having to store descriptive meta-data in the specification as full-blown items; these are just pseudo-items, and requiring entries in iloc, iinf and ipro for this is quite an overhead. Furthermore, this metadata is usually small and enables efficient reading of the other items. Having them stored as dedicated items may complicate file parsing, especially partial fetching of a file (multiplication of HTTP requests for example). We suggest defining embedded items which can be stored in the meta box hierarchy.

“

Virtual Items

A VirtualItemBox defines a virtual item. A virtual item has an item_ID and item_type assigned to it, along with a body, possibly consisting in set of boxes. Virtual items are typically used to describe metadata associated with other items. Virtual items may only be referenced in item reference boxes and primary item boxes, and shall not be declared or referenced in any other box (eg iloc, iinf, ipro).

```
aligned(8) class VirtualItemBox(unsigned int(32) item_type)
      extends FullBox('vite', version, 0) {

     if (version == 0) {
           unsigned int(16)  item_ID;
     } else {

          unsigned int(32)  item_ID;
     }
     unsigned int(32)  item_type;
}
```

Semantics :
`item_ID` **:** ID of this item. Like regular item_IDs, this ID shall be unique in the scope of the meta box. It is illegal to have entries in iinf, iloc or ipro of a meta at the same level (file, moov, track) with the same value of item_ID
`item_type` is a 32-bit value, typically 4 printable characters, that is a defined valid item type indicator, such as ‘dimg’.

”

For timing reasons, this modification can be included in the 23008-12, but should be introduced into 14496-12 as a generic tool for metadata and item descriptions.

We suggest changing the definitions of ImageOverlay, SubSampleItemData, AuxiliaryConfiguration, ExifDataBlock, ISOBMFFMetaData and derived image item to inherit from the virtual item class.

5 Recommendation

We recommend MPEG merging the approaches of the Sapporo and Strasbourg output by:
- As indicated in FRNB, introduce a single, generic item type called 'dimg', with item references of type 'simr' to items of types 'simd'This approach should enable the reuse of properties when appropriate and reduce the number of items and item references.
- Add the ImageRotationBox derived from the study into the ISOBMFFMetadata as described in section 2.1 of this contribution;
- Reverse the order of ‘simr’ reference so that an image item is linked to a ‘simd’ item, so as to provide direct access to image metadata.
- Revert ‘ispe’ (Study) to ‘isre’ (Sapporo) to enable sub-images position offsets for tiling description
- Introduce the concept of virtual items and move all descriptive items introduced in the IFF spec to this format.
- redesign the overlay so that it is not dependent on reference order. We suggest having an explicit item_id for each entry in the loop.