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

ISO/IEC JTC1/SC29/WG11 MPEG2014/M33221
March 2014, Valencia, Spain

Source **Telecom ParisTech, Canon Research Centre France**
Status **For consideration at the 108th MPEG Meeting**
Title **Storage of HEVC Tiled Images in the Image File Format**
Author Jean Le Feuvre, Cyril Concolato, Franck Denoual, Frédéric Mazé, Eric Nassor

1 Introduction

At the 107th Meeting, MPEG issued a CD for storage of HEVC still image in the ISO Base Media File Format [1]. This CD proposes technology to store single images, image collection or image sequences in ISOBMFF. However, the current CD does not propose any tool to describe and access independently coded tiles of an HEVC image. This contribution proposes a simple way to describe tiling in the HEVC still image format.

2 HEVC Image Sequences

The current CD covers two forms of storage, one relying on 'meta' boxes, the other one relying on 'trak' boxes. The second case allows HEVC image sequences to be stored as regular track in ISOBMFF, but with a media handler type 'pict'. In case tiling is used in the HEVC bitstream, such storage is compatible with the sample group descriptions currently defined in 14496-15 AMD1 WD2 [2].
The text however states:
"
If optional timing or other tools from the ISO Base Media File Format available for tracks are needed (e.g. a simple animated image), or pictures have been coded with inter-picture coding dependency, then the second approach is needed."

We believe the "(e.g. a simple animated image)" should be removed, and replaced with "(e.g. sample grouping)".

3 HEVC Still Image

The Still Image file format also allows storage of HEVC still pictures as items in the 'meta' box. The design is fairly simple, and mutualizes the HEVC parameter sets by using an item containing an 'hvcC' record, as defined in HEVC file format. All pictures sharing this record refer to it by means of entries in the item reference box having 'init' as reference type value. We propose to reuse and extend that design for tiles.

3.1 HEVC Still Image Tile Access

By design, meta items are not always stored contiguously in a file, and no restriction applies on how item data is interleaved; this also means that two items in a file could share one or several

blocks of data. This is particularly useful for HEVC tiles, since it is straightforward to have one item per independently decodable tiles, which will indicate data offset in the main HEVC picture and length of the slice(s) used for the tile through an ItemLocationBox. We therefore suggest adding a new item type "hvt1" which indicate that the item is an HEVC tile. Each "hvt1" item shall have a reference of type "tbas" to the 'hvc1' item it refers to; this reference implicitly indicates that it shares the same HEVCDecoderConfigurationRecord (no need to introduce an explicit 'init' reference).
This simple syntax allows a file reader to only download one tile of the image and the associated decoder config, while skipping the rest.
For cases where an HEVC tile depends on another HEVC tile for decoding, the dependency shall be indicated by an item reference of type 'dpnd' from [3].

3.2 HEVC Still Image and Tile data format

The current text should explicitly state that the HEVCItemData is structurally identical to the syntax defined in 14496-15.
Moreover, we suggest clarifying that data pointed to by an item of type 'hvc1' or 'hvt1' shall not contain any extractor or aggregator NAL units.

3.3 HEVC Still Image Tile Description

The HEVC file format WD [2] defines tools to associate HEVC tile NALUs to sample group descriptions indicating the spatial position of the tile (using for example the TileRegionGroupEntry descriptor). Unfortunately, there is no direct mapping of sample grouping for metadata items which could allow reuse of these descriptors.

We have investigated several options for describing tile data:

- Defining a single tile description item and linking the tile to its description through the ItemReferenceBox 'iref' ; this works well, however the solution has one drawback: the list of information items gets very big (one item per tile, one item per tile description) which is not really elegant
- Trying to use information on tiling from EXIF and reuse the mechanism defined in the CD [1], however we couldn't find EXIF tags allowing describing non-regular grid; moreover, the same drawback as above was found: too many items.

We therefore would like to define something similar to sample grouping for items, i.e. to have only one tiling description, and if possible define it in a generic way. The design of the solution is as follows:

- allow some items to describe a set of metadata, similar to sample groups but specific to each item type,
- for any item, add the ability to describe one parameter for a given type of item reference. The parameter would then be interpreted depending on the type of the referred item (similar to the `grouping_type` parameter).

This requires upgrading the 'infe' box as follows

Change

```
if (version == 2) {
 unsigned int(16) item_ID;
 unsigned int(16) item_protection_index;
 unsigned int(32) item_type;
```

```
 string item_name ;
 if (item_type==’mime’) {
  string  content_type;
  string  content_encoding; //optional
 } else if (item_type == ‘uri ‘) {
  string  item_uri_type;
 }
}
```

to

```
if ((version == 2) || (version == 3)) {
 unsigned int(16) item_ID;
 unsigned int(16) item_protection_index;
 unsigned int(32) item_type;

 string item_name;
 if (version == 2) {
  if (item_type==’mime’) {
   string content_type;
   string content_encoding; //optional
  } else if (item_type == ‘uri ‘) {
   string item_uri_type;
  }
 }
 if (version == 3) {
  unsigned int(32) item_iref_parameter_count;
  for (i=0 ; i< item_iref_parameter_count ; i++) {
   unsigned int(32) iref_type;
   unsigned int(32) iref_parameter;
  }
 }
}
```

`item_iref_parameter_count` gives the number of reference types for which a parameter is given

`iref_type` gives the reference type, as indicated in the ‘iref’ box, for which the parameter applies for this item

`iref_parameter` gives parameter associated to this item for the given `iref_type`. Semantics of the parameter is given by the semantics of the item with type `iref_type`. There may be several `iref_parameter` with the same `iref_type` indicated, depending on the semantics of the parameter.

This extension of `ItemInfoEntry` gives a generic framework allowing a reuse of items referenced through ‘iref’ box.

For describing the tiling, we propose to define the following item syntax

```
aligned(8) class TileInfoDataItem () {
 unsigned int(8) version;
 unsigned int(1) regular_spacing;  // regular grid or not
 unsigned int(7) reserved = 0;
 unsigned int(32) reference_width; // full-frame width
 unsigned int(32) reference_height; // full-frame height
 unsigned int(32) nb_cell_horiz;
 unsigned int(32) nb_cell_vert;
 if (!regular_spacing) {
  for (i=0; i<nb_cell_width; i++)
   unsigned int(16) cell_width;
```

```
  for (i=0; i<nb_cell_height; i++)
   unsigned int(16) cell_height;
  }
 }
}
```

The `TileInfoDataItem` allows describing a tiling grid, whether regular or irregular. The grid is described rows by rows starting from top-left.
The `TileInfoDataItem` shall be stored as an item of type 'tgif' (for tile grid information). When another item refers to this item, it shall use a reference of type "tgir" (for tile grid reference) to this description and it shall have a `iref_parameter` specified, whose value is the 0-based index of the cell in the grid defined by the `TileInfoDataItem`, where 0 is the top-left item, 1 is the cell immediately to the right of cell 0 and so on.

Semantics:
`version` indicates the version of the syntax for the `TileInfoDataItem`. Only value 0 is defined.
`regular_spacing` indicates if all tiles in the grid have the same width and the same height.
`reference_width, reference_height` indicate the units in which the grid is described. These units may or may not match the pixel resolution of the image which refers to this item. If the grid is regular, the `reference_width` (resp. `reference_height`) shall be a multiple of `nb_cell_horiz` (resp. `nb_cell_vert`).
`cell_width` gives the horizontal division of the grid in non-regular tiles, starting from the left.
`cell_height` gives the vertical division of the grid in non-regular tiles, starting from the top.

4 Example

```
ftyp box:   major-brand = 'hevc', compatible-brands = 'hevc'
meta box:   (container)
   handler box:   hdlr = 'hvc1'     // really redundant with the item type
   primary item: itemID = 1;
   Item information:
item_type = 'hvc1', itemID=1, item_protection_index = 0 (unused)
item_type = 'Exif', itemID=2, item_protection_index = 0 (unused)
item_type = 'hvcC', itemID=3, item_protection_index = 0 (unused)
item_type = 'hvt1', itemID=4, parameter for ireftype==tgir: tile_index=0
item_type = 'hvt1', itemID=5, parameter for ireftype==tgir: tile_index=1
item_type = 'hvt1', itemID=6, parameter for ireftype==tgir: tile_index=2
item_type = 'hvt1', itemID=7, parameter for ireftype==tgir: tile_index=3
item_type = 'tgif, itemID=8,

   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 = X, extent_length = ET1;
itemID = 5, extent_count = 1, extent_offset = X+ET1, extent_length = ET2;
itemID = 6, extent_count = 1, extent_offset = X+ET2, extent_length = ET3;
itemID = 7, extent_count = 1, extent_offset = X+ET3, extent_length = ET4;
itemID = 8, extent_count = 1, extent_offset = i, extent_length = I;

   Item Reference:
type='cdsc', fromID=2, toID=1;
type='init', fromID=1, toID=3;
type='tbas', fromID=4, toID=1;
type='tbas', fromID=5, toID=1;
type='tbas', fromID=6, toID=1;
type='tbas', fromID=7, toID=1;
```

```
type='tgir', fromID=4, toID=8;
type='tgir', fromID=5, toID=8;
type='tgir', fromID=6, toID=8;
type='tgir', fromID=7, toID=8;


Media data box:
   HEVC Image (at file offset X, with length Y)
   Exif data block (at file offset P, with length Q)
   HEVC Config Record (at file offset R, with length S)
   Tile description data block (at file offset i, with length I)
```

5 Recap of all 4CCs used in tiling for HEVC or for general media

Since many 4CC are being defined for tiling in all ISOBMFF drafts, we thought it could be useful to have an overview of these, and whether they are track references, item types or sample groups.

4CC	23008-12 (Still Image)	14496- 12 (m33222)	14496-15 WD2
hvt1	Item_type	N/A	Sample description type
tbas	Reference type (imply coding dep)	N/A	Reference type (imply coding dep)
tile	Could be defined	Reference type (no coding dep)	N/A
tgif	item_type	N/A	N/A
tgir	Reference type	N/A	N/A
trif / tsif	N/A	Sample group (tile indicates the full frame)	Sample group (tbas indicates the full frame)

6 Conclusion

We recommend MPEG to adopt the proposed modification and to include it in a study text of the CD on HEVC still image.

7 References

[1] w14148, Text of ISO/IEC CD 23008-12 Image File Format, MPEG 107 San José January 2014
[2] w14123, WD of ISO/IEC 14496-15:2013 AMD 1 Enhanced carriage of HEVC and support of MVC with depth information, MPEG 107 San José January 2014
[3] ISO/IEC 14496-14:2003, MP4 File Format, April 2003