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/M37250
October 2015, Geneva, Switzerland

Source	**Canon Research Centre France, Telecom ParisTech**
Status	**For consideration at the 113th MPEG Meeting**
Title	**Examples of use of the Image File Format**
Author	Franck Denoual, Frédéric Mazé, Cyril Concolato, Jean Le Feuvre

1 Introduction

During the 112th MPEG meeting, a new Text of ISO/IEC FDIS 23008-12 Carriage of Still Image and Image Sequences was issued with an informative Annex G providing usage examples. This contribution provides more examples of use trying to cover more exhaustively the Image File Format features. It also proposes some comments on current text.

2 Examples of single image and image collection

In the examples below, when parameters are omitted (to preserve readability) in boxes or items, they should be considered as equal to 0.

2.1 Single image use cases

2.1.1 Single Image

This use case is already provided in Annex G, but could be improved by:
- indicating mandatory property (‘ispe’)
- indicating the 4cc code for the boxes as follows and
- declaring parameters in order and with the actual order they are defined in their respective boxes.

The first example of Annex G, section G.2 could then be rewritten (highlighted parts) into:

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:    (container)
      HandlerBox ‘hdlr’:      'pict'
      PrimaryItemBox ‘pitm’: item_ID = 1;
      ItemInfoBox ‘iinf’: entry_count=2
            1)    ‘infe’: item_ID=1, item_type = 'hvc1';
            2)    ‘infe’: item_ID=2, item_type = 'Exif'

      ItemLocationBox ‘iloc’: item_count=2
            item_ID = 1, extent_count = 1, extent_offset = X, extent_length =
Y;
            item_ID = 2, extent_count = 1, extent_offset = P, extent_length =
Q;

      ItemReferenceBox ‘iref’:
            referenceType='cdsc', from_item_ID=2, reference_count=1,
to_item_ID=1;
```

```
      ItemPropertiesBox 'iprp':
            ItemPropertyContainerBox 'ipco':
                  'hvcC'
                  'ispe'
            ItemPropertyAssociation 'ipma': entry_count=1,
            1) item_ID=1, association_count=2
                  essential=1, property_index=1;
                  essential=0, property_index=2;

MediaDataBox 'mdat' or 'idat':
      HEVC Image (at file offset X, with length Y)
      Exif data block (at file offset P, with length Q)
```

Editorial Note on ISO/IEC 14496- 12: in section 8.11.6.2, the definition of 'infe' uses *item_ID* while in the semantics *item_id*.is used. It would be better to use *item_ID* everywhere. The same occurs in section 8.11.12.2 for the different kinds of ItemReferenceBox.

2.1.2 Single Image with its thumbnail

This example is a new one (not included in Annex G). It is the same as the simple image one plus the declaration of a thumbnail image. The thumbnail image is also an HEVC-coded image, but at smaller resoultion. (The 'Exif' item has been removed to focus on thumbnail declaration, but could of course be declared as in previous example).

```
FileTypeBox 'ftyp': major-brand = 'heic', compatible-brands = 'heic'
MetaBox 'meta':   (container)
      HandlerBox 'hdlr':      'pict'
      PrimaryItemBox 'pitm': item_ID = 1;
      ItemInfoBox 'iinf': entry_count=2
            1)    'infe': item_ID=1, item_type = 'hvc1'; // the master image
            2)    'infe': item_ID=2, item_type = ' hvc1'; // the thumbnail
image

      ItemLocationBox 'iloc': item_count=2
            item_ID = 1, extent_count = 1, extent_offset = X, extent_length =
Y;
            item_ID = 2, extent_count = 1, extent_offset = P, extent_length =
Q;

      ItemReferenceBox 'iref':
            referenceType='thmb', from_item_ID=2, reference_count=1,
to_item_ID=1;

      ItemPropertiesBox 'iprp':
            ItemPropertyContainerBox 'ipco':
                  'hvcC'      // Config for full image
                  'ispe'      // Sizes of full image
                  'hvcC'      // Config for thumbnail image
                  'ispe'      // sizes of thumbnail image
            ItemPropertyAssociation 'ipma': entry_count=2,
            1) item_ID = 1, association_count = 2
                  essential = 1, property_index = 1;
                  essential = 0, property_index = 2;
            2) item_ID = 2, association_count = 2
                  essential = 1, property_index = 3;
                  essential = 0, property_index = 4;
```

```
MediaDataBox ‘mdat’ or ‘idat’:
      HEVC Image (at file offset X, with length Y)
      HEVC smaller image (at file offset P, with length Q)
```

2.1.3 Single Image with an auxiliary image

This example (not included in Annex G.4) declares two HEVC coded images: one being an auxiliary image to the other one. This is reflected in an ItemReferenceBox and with a property assigned to the auxiliary image.
It is assumed in this example that the auxiliary image has the same HEVC configuration and dimensions as the master image

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: item_ID = 1;
   ItemInfoBox ‘iinf’:  entry_count=2,
      1) ‘infe’: item_ID=1, item_type = 'hvc1'; // the master image
      2) ‘infe’: item_ID=2, item_type = 'hvc1'; // the auxiliary image

   ItemLocationBox ‘iloc’: item_count=2,
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P, extent_length = Q;

   ItemReferenceBox ‘iref’:
      referenceType='auxl', from_item_ID=2, reference_count=1, to_item_ID=1;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’ // config shared between master and auxiliary images
         ‘ispe’ // same sizes for both master and auxiliary images
         ‘auxC’
      ItemPropertyAssociation ‘ipma’: entry_count=2,
         item_ID=1, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;
         item_ID=2, association_count=3,
            essential=1, property_index=1;
            essential=0, property_index=2;
            essential=0, property_index=3;

MediaDataBox:
   HEVC Master Image (at file offset X, with length Y)
   HEVC Auxiliary Image (at file offset P, with length Q)
```

2.1.4 Single Image with pre-derived coded image

This example is a modified (in yellow) version of the example in Annex G.4, including the 4cc codes of the boxes and the correct order and names of boxes parameters.
The primary item is a pre-derived coded image depending on the three other image items. All images are assumed to have the same width and height.

```
FileTypeBox 'ftyp':  major-brand = 'heic', compatible-brands = 'heic'
MetaBox 'meta':   (container)
   HandlerBox  'hdlr': 'pict'
   PrimaryItemBox 'pitm': item_ID = 1;
   ItemInfoBox 'iinf': entry_count=4
      1) 'infe': item_ID=1, item_type = 'hvc1';
      2) 'infe': item_ID=2, item_type = 'hvc1';
      3) 'infe': item_ID=3, item_type = 'hvc1';
      4) 'infe': item_ID=4, item_type = 'hvc1';

   ItemLocationBox 'iloc': item_count=4
      item_ID = 1, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 2, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 3, extent_count = 1, extent_offset = P2, extent_length = Q2;
      item_ID = 4, extent_count = 1, extent_offset = P3, extent_length = Q3;

   ItemReferenceBox:
      referenceType='base', from_item_ID=1, reference_count=3,
         to_item_ID=2,
         to_item_ID=3,
         to_item_ID=4;

   ItemPropertiesBox 'iprp':
      ItemPropertyContainerBox 'ipco':
         'hvcC'
         'ispe'
      ItemPropertyAssociation 'ipma': entry_count=4
         1) item_ID = 1, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;
         2) item_ID = 2, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;
         3) item_ID = 3, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;
         4) item_ID = 4, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;

MediaDataBox:
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
   HEVC Image (at file offset P3, with length Q3)
```

2.1.5 Single Image with rotation

This example is a new one (not included in Annex G). It describes a rotated image.

```
FileTypeBox 'ftyp': major-brand = 'heic', compatible-brands = 'heic'
MetaBox 'meta':   (container)
   HandlerBox 'hdlr':   'pict'
   PrimaryItemBox 'pitm': item_ID = 1;
   ItemInfoBox ‘iinf’: entry_count=1
      1) ‘infe’: item_ID=1, item_type = 'hvc1';

   ItemLocationBox ‘iloc’: item_count=1
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
         ‘irot’
      ItemPropertyAssociation ‘ipma’: entry_count=1,
         1) item_ID=1, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=3;

MediaDataBox ‘mdat’ or ‘idat’:
      HEVC Image (at file offset X, with length Y)
```

2.1.6 Single Image and its cropped version

This example is a new one (not included in Annex G). It describes the source image and a cropped version of this source image through a derived image (corresponds to the note of section 6.6.2.1). The primary item is the cropped version.

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox ‘hdlr’:   'pict'
   PrimaryItemBox ‘pitm’: item_ID = 1;

   ItemInfoBox ‘iinf’: entry_count=2
      1) ‘infe’: item_ID=1, item_protection_index = 0, item_type = 'hvc1',
item_name=0;

   ItemLocationBox ‘iloc’: item_count=1
      item_ID = 1, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = X, extent_length = Y;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
         ‘clap’
      ItemPropertyAssociation ‘ipma’: entry_count=1,
         1) item_ID=1, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=3;
```

```
MediaDataBox ‘mdat’ or ‘idat’:
   HEVC Image (at file offset X, with length Y)
```

2.1.7 Single “crop-and-rotate” Image

This example is a new one (not included in Annex G). The source image is an HEVC-coded image that is **cropped and then rotated** before being displayed (as specified by section 6.3). The so-obtained derived image is declared as an item that is also the primary item.

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox ‘hdlr’:   'pict'
   PrimaryItemBox ‘pitm’: item_ID = 1;
   ItemInfoBox ‘iinf’: entry_count=1
      1) ‘infe’: item_ID=1, item_type = 'hvc1';


   ItemLocationBox ‘iloc’: item_count=1
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
         ‘clap’
         ‘irot’
      ItemPropertyAssociation ‘ipma’: entry_count=1,
         1) item_ID=1, association_count=4,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=3;
               essential=0, property_index=4;

MediaDataBox ‘mdat’ or ‘idat’:
      HEVC Image (at file offset X, with length Y)
```

2.1.8 Single Image with its tiles

This example is a new one (not included in Annex G) showing how to declare an HEVC-coded image that is a tiled HEVC picture containing 4 tiles. Each tile is an item, containing position information in ‘rloc’. This requires an ItemReferenceBox to declare the relation between tiles and their source image.

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox ‘hdlr’:   'pict'
   PrimaryItemBox ‘pitm’: item_ID = 1;
   ItemInfoBox ‘iinf’: entry_count=1
      1) ‘infe’: item_ID=1, item_type = 'hvc1'; // full image
      2)    ‘infe’: item_ID=2, item_type = 'hvt1'; // first tile
      3)    ‘infe’: item_ID=3, item_type = 'hvt1'; // second tile
      4)    ‘infe’: item_ID=4, item_type = 'hvt1'; // third tile
      5)    ‘infe’: item_ID=5, item_type = 'hvt1'; // fourth tile
   ItemLocationBox ‘iloc’: item_count=4
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length =
Q0+Q1+Q2+Q3;
      item_ID = 2, extent_count = 1, extent_offset = X, extent_length = Q0;
      item_ID = 1, extent_count = 1, extent_offset = X+Q0, extent_length =
Q1;
      item_ID = 1, extent_count = 1, extent_offset = X+Q0+Q1, extent_length =
Q2;
      item_ID = 1, extent_count = 1, extent_offset = X+Q0+Q1+Q2,
extent_length = Q3;

   ItemReferenceBox ‘iref’:
      reference_type='tbas', from_item_ID=2, reference_count=1, to_item_ID=1;
      reference_type='tbas', from_item_ID=3, reference_count=1, to_item_ID=1;
      reference_type='tbas', from_item_ID=4, reference_count=1, to_item_ID=1;
      reference_type='tbas', from_item_ID=5, reference_count=1, to_item_ID=1;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’  // size of the full image
         ‘rloc’  // position for 1st tile
         ‘rloc’  // position of 2nd tile
         ‘rloc’  // position of 3rd tile
         ‘rloc’  // position of 4th tile
         ‘ispe’  // Sizes of the tiles (assuming all have same sizes)

      ItemPropertyAssociation ‘ipma’: entry_count=5, // full image + 4 tiles
         1) item_ID=1, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
         2) item_ID=2, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=3;
               essential=0, property_index=7;
         3) item_ID=3, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=4;
               essential=0, property_index=7;
         4) item_ID=4, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=5;
               essential=0, property_index=7;
         5) item_ID=5, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=6;
               essential=0, property_index=7;
```

```
MediaDataBox ‘mdat’ or ‘idat’:
   HEVC Image (at file offset X, with length Q0+Q1+Q2+Q3)
```

2.1.9 Single Image with a region of interest (as a tile)

This example is a new one (not included in Annex G) showing how to declare a region of interest in an HEVC-coded image. The region of interest, corresponding to a tile, is the primary item. This could be used for storing the result of art direction as used in responsive design of web sites.

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox ‘hdlr’:   'pict'
   PrimaryItemBox ‘pitm’: item_ID = 1;
   ItemInfoBox ‘iinf’: entry_count=1
      1) ‘infe’: item_ID=1, item_type = 'hvt1';  // The region of interest
      2)    ‘infe’: item_ID=2, item_type = 'hvc1';  // The full image.

   ItemLocationBox ‘iloc’: item_count=2
      item_ID = 1, extent_count = 1, extent_offset = X+Q0, extent_length = 
Q1;
      item_ID = 2, extent_count = 1, extent_offset = X, extent_length = 
Q0+Q1+Q2;

   ItemReferenceBox ‘iref’:
      referenceType='tbas', from_item_ID=1, reference_count=1, to_item_ID=2;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’   // size of the full image
         ‘ispe’   // size of the ROI
         ‘rloc’   // position of the ROI
      ItemPropertyAssociation ‘ipma’: entry_count=2, // full image + ROI
         1) item_ID=1, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=3;
               essential=0, property_index=4;
         2) item_ID=2, association_count=2,
               essential=1, property_index=1;
               essential=0, property_index=2;

MediaDataBox ‘mdat’ or ‘idat’:
   HEVC Image (at file offset X, with length Q0+Q1+Q2)
```

2.1.10 Single Image sharing HEVC config with a track (dual-function file as in G.4)

The example has been edited to highlight the 4cc codes of the boxes and use correct parameters names.

```
FileTypeBox 'ftyp':  major-brand = 'heic', compatible-brands = 'heic, mp41'
MetaBox: (container)
   HandlerBox 'hdlr': 'pict'
   PrimaryItemBox 'pitm': item_ID = 1;
   ItemInfoBox 'iinf':  entry_count=2
      1) 'infe': item_type = 'hvc1', item_ID=1;
      2) 'infe': item_type = 'Exif', item_ID=2;

   ItemLocationBox'iloc': item_count=2
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P, extent_length = Q;
   ItemReferenceBox 'iref':
      referenceType ='cdsc', from_item_ID=2, reference_count=1, to_item_ID=1;

   ItemPropertiesBox 'iprp':
      ItemPropertyContainerBox 'ipco':
         'hvcC'
         'ispe'
      ItemPropertyAssociation 'ipma':
         item_ID=1, association_count=2,
            essential=1, property_index=1;
            essential=0, property_index=2;

Movie Box 'moov': (container)
   Movie header, tracks, etc. as required by MP4
MediaDataBox 'mdat':
   HEVC Image (at file offset X, with length Y)
   Exif data block (at file offset P, with length Q)
   Media data as needed by the movie (some may be shared with the image data)
```

2.2 Multiple Image use cases

2.2.1 Multiple Images with the same dimensions but different HEVC configurations

This example with 4 HEVC coded images is a new one (not included in Annex G). Each coded image has its own HEVC configuration and sizes.

```
FileTypeBox 'ftyp':  major-brand = 'heic', compatible-brands = 'heic'
MetaBox 'meta':    (container)
   HandlerBox 'hdlr': 'pict'
   PrimaryItemBox 'pitm': itemID = 1;
   ItemInfoBox 'iinf':  entry_count=4
      1) 'infe': item_ID=1, item_type = 'hvc1';
      2) 'infe': item_ID=2, item_type = 'hvc1';
      3) 'infe': item_ID=3, item_type = 'hvc1';
      4) 'infe': item_ID=4, item_type = 'hvc1';

   ItemLocationBox 'iloc': item_count=4
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;

   ItemPropertiesBox 'iprp':
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘hvcC’
         ‘hvcC’
         ‘hvcC’
         ‘ispe’
         ‘ispe’
         ‘ispe’
         ‘ispe’
      ItemPropertyAssociation ‘ipma’: entry_count=4,
         1) item_ID=1, association_count=2,
               essential=1, property_index=1;
               essential=0, property_index=5;
         2) item_ID=2, association_count=2,
               essential=1, property_index=2;
               essential=0, property_index=6
         3) item_ID=3, association_count=2,
               essential=1, property_index=3;
               essential=0, property_index=7;
         4) item_ID=4, association_count=2,
               essential=1, property_index=4;
               essential=0, property_index=8;

   MediaDataBox ‘mdat’ or ‘idat’:
      HEVC Image (at file offset X, with length Y)
      HEVC Image (at file offset P0, with length Q0)
      HEVC Image (at file offset P1, with length Q1)
      HEVC Image (at file offset P2, with length Q2)
```

2.2.2 Multiple Images, each with its thumbnail

This example is a new one (not included in Annex G). It is the same as the simple image one plus the declaration of a thumbnail image but extended to 2 images here. The thumbnail images are also HEVC-coded images, but at smaller resolution. (The ‘Exif’ item has been removed to focus on thumbnail declaration, but could of course be declared as in previous example).

```
FileTypeBox ‘ftyp’: major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
      HandlerBox ‘hdlr’:      'pict'
      PrimaryItemBox ‘pitm’: item_ID = 1;
      ItemInfoBox ‘iinf’: entry_count=4
            1)    ‘infe’: item_ID=1, item_type='hvc1'; // 1st image
            2)    ‘infe’: item_ID=2, item_protection_index=0,
item_type='hvc1'; // 1st thumbnail
            3)    ‘infe’: item_ID=3, item_type='hvc1'; // 2nd image
            4)    ‘infe’: item_ID=4, item_type='hvc1'; // 2nd thumbnail

      ItemLocationBox ‘iloc’: item_count=4
            item_ID = 1, extent_count = 1, extent_offset = X1, extent_length
= Y1;
            item_ID = 2, extent_count = 1, extent_offset = P1, extent_length
= Q1;
            item_ID = 3, extent_count = 1, extent_offset = X2, extent_length
= Y2;
```

```
          item_ID = 4, extent_count = 1, extent_offset = P2, extent_length
= Q2;

      ItemReferenceBox ‘iref’:
            referenceType='thmb', from_item_ID=2, reference_count=1,
to_item_ID=1;

      ItemReferenceBox ‘iref’:
            referenceType='thmb', from_item_ID=4, reference_count=1,
to_item_ID=3;

      ItemPropertiesBox ‘iprp’:
            ItemPropertyContainerBox ‘ipco’:
                  ‘hvcC’      // Config for full image
                  ‘ispe’      // Sizes of full image
                  ‘hvcC’      // Config for thumbnail image
                  ‘ispe’      // sizes of thumbnail image
            ItemPropertyAssociation ‘ipma’: entry_count=4,
               1) item_ID = 1, association_count=2
                  essential = 1, property_index = 1;
                  essential = 0, property_index = 2;
               2) item_ID = 2, association_count=2
                  essential = 1, property_index = 3;
                  essential = 0, property_index = 4;
               3) item_ID = 3, association_count=2
                  essential = 1, property_index = 1;
                  essential = 0, property_index = 2;
               4) item_ID = 4, association_count=2
                  essential = 1, property_index = 3;
                  essential = 0, property_index = 4;


MediaDataBox ‘mdat’ or ‘idat’:
      HEVC First Image (at file offset X1, with length Y1)
      HEVC smaller image (at file offset P1, with length Q1)
      HEVC Second Image (at file offset X2, with length Y2)
      HEVC smaller image (at file offset P2, with length Q2)
```

2.2.3 Multiple Images with the same HEVC configuration and dimensions

This example with 4 HEVC coded images is a new one (not included in Annex G).

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 1;
   ItemInfoBox ‘iinf’:  entry_count=4
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';

   ItemLocationBox ‘iloc’: item_count=4
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;
   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
      ItemPropertyAssociation ‘ipma’: entry_count=4,
         1) item_ID=1, association_count=2,
               essential=1, property_index=1
               essential=0, property_index=2
         2) item_ID=2, association_count=2,
               essential=1, property_index=1
               essential=0, property_index=2
         3) item_ID=3, association_count=2,
               essential=1, property_index=1
               essential=0, property_index=2
         4) item_ID=4, association_count=2,
               essential=1, property_index=1
               essential=0, property_index=2

   MediaDataBox ‘mdat’ or ‘idat’:
      HEVC Image (at file offset X, with length Y)
      HEVC Image (at file offset P0, with length Q0)
      HEVC Image (at file offset P1, with length Q1)
      HEVC Image (at file offset P2, with length Q2)
```

2.2.4 Multiple Images with the same HEVC configuration and dimensions but different rotations

This example with 4 HEVC coded images is a new one (not included in Annex G). 4 HEVC coded images are declared and each have different rotation angles.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 1;
   ItemInfoBox ‘iinf’:  entry_count=4
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';

   ItemLocationBox ‘iloc’: item_count=4
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
         ‘irot’
         ‘irot’
         ‘irot’
      ItemPropertyAssociation ‘ipma’: entry_count=4,
         1) item_ID=1, association_count=2,
               essential=1, property_index=1;
               essential=0, property_index=2;
         2) item_ID=2, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=3;
         3) item_ID=3, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=4;
         4) item_ID=4, association_count=3,
               essential=1, property_index=1;
               essential=0, property_index=2;
               essential=0, property_index=5;

MediaDataBox:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
```

2.2.5 Multiple Images in a grid

This example with 4 HEVC coded images is a new one (not included in Annex G).

A derived image is formed as the composition of 4 independent HEVC coded sub-images into a larger one. All images, including the derived one, are exposed as displayable items. An ItemReferenceBox is then used to declare the ‘grid’ derived image and its source images.

The primary item corresponds to this derived image, the 4 source images share the same size and configuration information.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 5;
   ItemInfoBox ‘iinf’:  entry_count=5
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';
      5) ‘infe’: item_ID=5, item_type = 'grid';

   ItemLocationBox ‘iloc’: item_count=5 // 4 source image items + grid info
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;
      item_ID = 5, extent_count = 1, extent_offset = P3, extent_length = Q3;


   ItemReferenceBox ‘iref’: // from which image the grid is built
      referenceType='dimg', from_item_ID=5, reference_count=4,
            to_item_ID=1;
            to_item_ID=2;
            to_item_ID=3;
            to_item_ID=4;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’ // all source image items share the same configuration info
         ‘ispe’ // all source image items shall have the same width/height.
      ItemPropertyAssociation ‘ipma’: entry_count=4,
         1) item_ID=1, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2
         2) item_ID=2, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2
         3) item_ID=3, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2
         4) item_ID=4, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2

MediaDataBox:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
   Grid parameters (at file offset P3, with length Q3)
```

2.2.6 Multiple different Images in a grid

This example with 4 HEVC coded images is a bit different than the above one, because here the sub-images involved in the composition are not all compatible in terms of configuration information. Indeed, the first two sub-image pictures (items 1 and 2) share common initialization data while the two other sub image pictures (items 3 and 4) share other initialization data.
All images, including the derived one, are exposed as displayable items. An ItemReferenceBox is then used to declare the ‘grid’ derived image and its source images. The primary item corresponds to this derived image.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 5;
   ItemInfoBox ‘iinf’:  entry_count=5
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';
      5) ‘infe’: item_ID=5, item_type = 'grid';

   ItemLocationBox ‘iloc’: item_count=5 // 4 source image items + grid info
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;
      item_ID = 5, extent_count = 1, extent_offset = P3, extent_length = Q3;

   ItemReferenceBox ‘iref’: // from which image the grid is built
      referenceType='dimg', from_item_ID=5, reference_count=4,
            to_item_ID=1;
            to_item_ID=2;
            to_item_ID=3;
            to_item_ID=4;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’ // Configuration info for 2 first sub-images
         ‘ispe’ // Width/height for the 2 first sub-images
         ‘hvcC’ // Configuration info for 2 last sub-images
         ‘ispe’ // Width/height for the 2 last sub-images

      ItemPropertyAssociation ‘ipma’: entry_count=4,
         5) item_ID=1, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2
         6) item_ID=2, association_count=2,
               essential=1 , property_index=1
               essential=0 , property_index=2
         7) item_ID=3, association_count=2,
               essential=1 , property_index=3
               essential=0 , property_index=4
         8) item_ID=4, association_count=2,
               essential=1 , property_index=3
               essential=0 , property_index=4

MediaDataBox:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
   Grid parameters (at file offset P3, with length Q3)
```

2.2.7 Multiple rotated Images in a grid

This example with 4 HEVC coded images is a new one (not included in Annex G).

A derived image is formed as the composition of 4 rotated (all 90°) independent HEVC coded sub-images.

The primary item corresponds to this derived image, the 4 source images share the same size and configuration information.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 5;
   ItemInfoBox ‘iinf’:  entry_count=5
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';
      5) ‘infe’: item_ID=5, item_type = 'grid';

   ItemLocationBox ‘iloc’: item_count=5 // 4 source image items + grid info
      item_ID = 1, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, extent_count = 1, extent_offset = P2, extent_length = Q2;
      item_ID = 5, extent_count = 1, extent_offset = P3, extent_length = Q3;

   ItemReferenceBox ‘iref’:
      referenceType='dimg', from_item_ID=5, reference_count=4,
            to_item_ID=1;
            to_item_ID=2;
            to_item_ID=3;
            to_item_ID=4;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’
         ‘irot’
      ItemPropertyAssociation ‘ipma’: entry_count=4,
         1) item_ID=1, association_count=3,
               essential=1 , property_index=1
               essential=0 , property_index=2
               essential=0 , property_index=3
         2) item_ID=2, association_count=3,
               essential=1 , property_index=1
               essential=0 , property_index=2
               essential=0 , property_index=3
         3) item_ID=3, association_count=3,
               essential=1 , property_index=1
               essential=0 , property_index=2
               essential=0 , property_index=3
         4) item_ID=4, association_count=3,
               essential=1 , property_index=1
```

```
                essential=0 , property_index=2
                essential=0 , property_index=3

MediaDataBox:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
   Grid parameters (at file offset P3, with length Q3)
```

2.2.8 Multiple Images in a rotated grid

This example with 4 HEVC coded images is a new one (not included in Annex G).
A derived image is formed as the composition of 4 HEVC coded sub-images sharing the same configuration information and having the same sizes.
The primary item corresponds to this derived image rotated by 90°. This requires 5 items to be declared.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 5;
   ItemInfoBox ‘iinf’:  entry_count=5
      1) ‘infe’: item_ID=1, item_type = 'hvc1';
      2) ‘infe’: item_ID=2, item_type = 'hvc1';
      3) ‘infe’: item_ID=3, item_type = 'hvc1';
      4) ‘infe’: item_ID=4, item_type = 'hvc1';
      5) ‘infe’: item_ID=5, item_protection_index = 0, item_type = 'grid',
item_name=0; // the grid with 4 previous images



   ItemLocationBox ‘iloc’: item_count=5 // 4 source image items + grid info
      item_ID = 1, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P0, extent_length = Q0;
      item_ID = 3, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 4, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P2, extent_length = Q2;
      item_ID = 5, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P3, extent_length = Q3;

   ItemReferenceBox ‘iref’: // link from grid to the source images
      referenceType='dimg', from_item_ID=5, reference_count=4,
            to_item_ID=1;
            to_item_ID=2;
            to_item_ID=3;
            to_item_ID=4;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
```

```
          ‘ispe’ // for the source images
          ‘ispe’ // for the derived image
          ‘irot’
       ItemPropertyAssociation ‘ipma’: entry_count=4,
          1) item_ID=1, association_count=2,
                essential=1 , property_index=1;
                essential=0 , property_index=2;
          2) item_ID=2, association_count=2,
                essential=1 , property_index=1;
                essential=0 , property_index=2;
          3) item_ID=3, association_count=3,
                essential=1 , property_index=1;
                essential=0 , property_index=2;
          4) item_ID=4, association_count=3,
                essential=1 , property_index=1;
                essential=0 , property_index=2;
          5) item_ID=5, association_count=2,
                essential=0 , property_index=3;
                essential=0 , property_index=4;

MediaDataBox ‘mdat’:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P0, with length Q0)
   HEVC Image (at file offset P1, with length Q1)
   HEVC Image (at file offset P2, with length Q2)
   Grid parameters (at file offset P3, with length Q3)
```

2.2.9 Multiple Images with overlay

This example provides an example of overlay derived image from two input images.

```
FileTypeBox ‘ftyp’:  major-brand = 'heic', compatible-brands = 'heic'
MetaBox ‘meta’:   (container)
   HandlerBox  ‘hdlr’: 'pict'
   PrimaryItemBox ‘pitm’: itemID = 3;
   ItemInfoBox ‘iinf’: entry_count=6
      1) ‘infe’: itemID=1, item_protection_index = 0, item_type = 'hvc1',
item_name=0;
      2) ‘infe’: itemID=2, item_protection_index = 0, item_type = 'hvc1',
item_name=0;
      3) ‘infe’: itemID=3, item_protection_index = 0, item_type = 'iovl',
item_name=0;

   ItemLocationBox ‘iloc’: item_count=3   // 2 images + overlay info
      item_ID = 1, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = X, extent_length = Y;
      item_ID = 2, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P1, extent_length = Q1;
      item_ID = 3, construction_method = 0, data_reference_index =0,
base_offset=0, extent_count = 1, extent_offset = P2, extent_length = Q2;

   ItemReferenceBox ‘iref’:
      referenceType='dimg', from_item_ID=3, reference_count=2,
         to_item_ID=1,
         to_item_ID=2;

   ItemPropertiesBox ‘iprp’:
      ItemPropertyContainerBox ‘ipco’:
         ‘hvcC’
         ‘ispe’// for the source images
         ‘ispe’// for the derived image
      ItemPropertyAssociation ‘ipma’: entry_count=2,
         1) item_ID=1, association_count=2,
               essential=1 , property_index=1;
               essential=0 , property_index=2;
         2) item_ID=2, association_count=2,
               essential=1 , property_index=1;
               essential=0 , property_index=3;

MediaDataBox:
   HEVC Image (at file offset X, with length Y)
   HEVC Image (at file offset P1, with length Q1)
   Overlay parameters (at file offset P2, with length Q2)
```

3 Comments on current text

3.1 Proposal

There are many cases where in ItemReferenceBox we have patterns like:
One or more items having a single reference of the same given type to the same item.
Then, why not defining a new kind of ItemTypeReference like the one below, to save some reference_count and to_item_ID parameters?

```
aligned(8) class SharedItemTypeReferenceBox(ref_type) extends
Box(referenceType) {
   unsigned int(16) reference_count;
   for (j=0; j<reference_count; j++) {
      unsigned int(16) from_item_ID;
   }
   unsigned int(16) to_item_ID;
   }
}
```

The box “ItemReferenceBox” would also need to be extended by distinguishing between the various kinds of ItemReferenceBox either by using the flag parameters in the box “FullBox” which is part of the ItemReferenceBox as follows, or using a new version of the box, as follows:

```
aligned(8) class ItemReferenceBox extends FullBox(‘iref’, version, 0) {
      if (version==0) {
            SingleItemTypeReferenceBox          references[];
      } else if (version==1) {
            SingleItemTypeReferenceBoxLarge     references[];
      } else if (version==2) {
            SharedItemTypeReferenceBox          references[];
```

```
        }
}
```

3.2 *Questions on current spec*

3.2.1 Section A.1.2

Shouldn't the IFF brands specify the name and version of all boxes required for a reader (infe, ipco, ipma, etc) ? And for conforming files ? These two tables seem uncomplete.

3.2.2 Section B.2.2 on (HEVC) Image data

There shall be no inter prediction between HEVC image items.
In case of multi-layer HEVC still image, what is the HEVC image item?
- All the layers?
- one layer?
- an ouput layer set ?

3.2.3 Section B.2.3 on HEVC configuration item property

Could we add that an 'hvcC' item property can be shared among several 'hvc1' items?

3.2.4 Section B.2.6 on HEVC Tile Items

It would be nice to recall that tiles shall have a tbas reference to the 'hvc1' item they refere to, if present. We suggest replacing

"Each HEVC tile item shall be associated with one `HEVCConfigurationBox`, one `ImageSpatialExtentsProperty` and one `RelativeLocationProperty`.
"

with

"Each HEVC tile item shall be associated with one `HEVCConfigurationBox`, one `ImageSpatialExtentsProperty` and one `RelativeLocationProperty`, and shall have a reference of type `'tbas'` from the tile item to the source image, as documented in section 6.5.7.1:"

4 Conclusion

We recommend MPEG to insert the provided examples of use in the Annex of the DIS text or in guidelines or white paper document, and to start an amendment with the proposed text improvements.