1 Introduction

During the 104th MPEG Meeting in Incheon, we presented a way to store and document HEVC tiles in ISOBMFF, as discussed in contribution m29231. We defined:
- a sample group describing the NALU structure of each sample in terms of tiles (named TileNALUMapEntry),
- a sample group describing spatial areas of the HEVC frame (named Tile Region Sample Group Entry) to declare each tile position and coding dependencies,
- and a specific use of extractors for HEVC with the notion of HEVC sub-layers.

After presentation to the group, it was decided that further investigations were needed, especially in the usage of sub-tracks for HEVC.

In this contribution, we refine and modify our previous proposal to further align with existing tools as much as possible, including with the notion of sub-tracks.

Additionally, in related contribution m30294, we propose a complementary mechanism which can be used if HEVC tiles need to be stored in separate tracks.

2 Proposal

2.1.1 Generic mapping of NAL Units to Map groups

In m29231, we proposed the TileNALUMapEntry. Following the group recommendations, we investigated the use of the ScalableNALUMapEntry and report them in this section.

The ScalableNALUMapEntry could be used in some cases in place of the proposed TileNALUMapEntry, especially when the tiling grid is not too fine (i.e with less than 256 tiles). For other cases, we believe a NALU partition box that allows more than 255 different labels is needed because this is not possible with the 8 bits used in the ScalableNALUMapEntry. Since the ScalableNALUMapEntry is not an extensible box, we therefore suggest using the box proposed in m29231, with a generic naming for all NALU-based videos.

We propose to define a generic NALU map entry as follows (with optional run-length encoding of the mapping):

```plaintext
class NALUMapEntry() extends VisualSampleGroupEntry ('nalm') {
```
unsigned int(6) reserved = 0;
unsigned int(1) large_size;
unsigned int(1) rle;
if (large_size) {
    unsigned int(16) entry_count;
} else {
    unsigned int(8) entry_count;
}
for (i=1; i<= entry_count; i++)
if (rle) {
    if (large_size) {
        unsigned int(16) NALU_start_number;
    } else {
        unsigned int(8) NALU_start_number;
    }
}
unsigned int(16) groupID;
}

**Semantics**

large_size indicates whether the number of NAL units entries in the track samples is represented on 8 or 16 bits.

rle indicates whether run-length encoding is used (1) to assign tile identifiers to NAL units or not (0).

entry_count specifies the number of entries in the map. Note that when rle is equal to 1, the entry_count corresponds to the number of runs where consecutives NAL units are associated with the same tile. When rle is equal to 0, entry_count represents the total number of NAL units.

NALU_start_number is the 1-based NALU index in the sample of the first NALU in the current run associated with groupID.

groupID indicates the corresponding scalable, multiview or tile group entry, as indicated in the sample group descriptions. If 0, no group is associated to these identified NALUs.

When used with movie fragments, new NALU maps can be defined for the duration of the movie fragments, by defining a new SampleGroupDescription box in the track fragment, box as defined in clause 8.9.4 of ISO/IEC 14496-12.

### 2.1.2 Specific group entry for tiles

In m29231, we proposed a tile group entry. We investigated its similarity with the RectRegionBox. We could use the RectRegionBox to describe the position, but this would require defining tierID for tiles, which is an unneeded complexity in our opinion. Therefore, we believe the TileRegionGroupEntry is still required. We propose to define it as described below, changing the tileID to groupID, and the 4CC code from trsg to trif in order to harmonize the terminology (with ScalableGroupEntry or MultiviewGroupEntry).

class TileRegionGroupEntry() extends VisualSampleGroupEntry ('trif')
{
    unsigned int(16) groupID;
    unsigned int(2) independent;
    unsigned int(6) reserved=0;
    unsigned int(16) horizontal_offset;
    unsigned int(16) vertical_offset;
    unsigned int(16) region_width;
    unsigned int(16) region_height;

    //other boxes
Semantics

`groupID` is a unique identifier for the tile described by this group. Value 0 is reserved for special use in the 'nalm' box.

`independent` specifies the coding dependencies between this tile and other tiles in the current frame and in reference frames:
- If independent equals 0, the coding dependencies between this tile and other tiles in the same frame or previous frames is unknown,
- If independent equals 1, there are no spatial coding dependencies between this tile and other tiles in the same frame, no temporal dependencies between this tile and the other tiles with different `groupID` in any reference frames but there can be coding dependencies between this tile and the tile with the same `groupID` in the reference frames,
- If independent equals 2, there are no coding dependencies between this tile and other tiles in the same frame, and no coding dependencies between this tile and any other tiles in the reference frames,
- Value 3 is reserved.

`horizontal_offset` and `vertical_offset` give respectively the horizontal and vertical offsets of the top-left pixel of the rectangular region represented by the tile, relative to the top-left pixel of the frame, in luma samples of the base region.

`region_width` and `region_height` give respectively the width and height of the rectangular region represented by the tile, in luma samples of the frame.

When used with movie fragments, new `TileRegionGroupEntry` can be defined for the duration of the movie fragments, by defining a new `SampleGroupDescription` box in the track fragment, box as defined in clause 8.9.4 of ISO/IEC 14496-12. However, there shall not be any `TileRegionGroupEntry` in a track fragment that has the same `groupID` as a `TileRegionGroupEntry` already defined in the track.

2.1.3 Tile description with sub tracks

In m29231, we introduced the concept of HEVC sub-layer. We report here our investigation relative to the harmonization with the existing concept of sub track.

Annex B.6 of ISO/IEC 14496-15:2013 defines sub tracks for SVC and MVC. We introduce here a sub track for HEVC and a specific sub track for tiles in HEVC. Sub tracks are defined using 2 boxes: `SubTrackInformationBox` and `SubTrackDefinitionBox`.

2.1.3.1 Sub track information box (section 8.14.4)

To support tile description with sub tracks, we extend the `attribute_list` of the `SubTrackInformationBox` with this additional descriptive attribute (section 8.14.4.3 of ISO/IEC 14496-12):

<table>
<thead>
<tr>
<th>Name</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiling</td>
<td>‘tile’</td>
<td>The sub-track is spatial part or tile of the track.</td>
</tr>
</tbody>
</table>

2.1.3.2 Sub track definition box (section 8.14.5)

The `SubTrackDefinitionBox` is a container for format specific sub tracks (ex: `SVCSubTrackLayerBox` or `MVCSubTrackViewBox` ...) and for optional sample grouping information `SubtrackSampleGroupBox` (‘stsg’ in section 8.14.6) as shown below:

```c
aligned(8) class SubTrackSampleGroupBox extends FullBox('stsg', 0, 0){
    unsigned int(32) grouping_type;
    unsigned int(16) item_count;
    for(i = 0; i < item_count; i++)
        unsigned int(32) group_description_index;
}
```
In the particular case of tile sub tracks, the `SubtrackSampleGroupBox` indicates:
- as `grouping_type` the ‘trif’ value to indicate a `TileRegionGroupEntry`
- as `group_description_index` the index of the `TileRegionGroupEntry` for the current tile.

Additional grouping information can be provided to define a sub track as a group of tiles. We define the `TileSubTrackGroupBox` as follows:

**Tile Sub Track Group box**

**Definition**
- Box Type: ‘tstb’
- Container: Sub Track Definition box (‘strd’)
- Mandatory: No
- Quantity: Zero or more

**Syntax**
```aligned(8)
class TileSubTrackGroupBox extends FullBox('tstb', 0, 0) {
    unsigned int(16) item_count;
    for(i = 0; i < item_count; i++) {
        unsigned int(16) GroupId;
    }
}
```

**Semantics**
The union of GroupIds in this box describes the sub track defined by this box.
- `item_count` counts the number of tile groups listed in this box.
- `GroupId` is the identifier of the tile group contained in this sub track.

This box can be used to describe a sub track corresponding to a pre-defined region of interest that would cover more than one tile. In this case, the `SubTrackDefinitionBox` would contain a `TileSubTrackGroupBox` to indicate which tiles (through the list of `groupId`) are covered by the pre-defined ROI associated to the sub track.

### 3 Conclusion

In this contribution, we defined a set of tools used to classify HEVC tiles into sub tracks, as well as a generic NALUMap entry to associate NALUs with sample group descriptions, valid for all NAL-based video coding schemes. Although the tools are defined with HEVC in mind, they are designed in a codec-agnostic way and could be reused for any codec providing a coding construct similar to HEVC tiles.

We recommend MPEG to adopt the definition of the NALUMapEntry and `TileRegionGroupEntry` and to consider the definition of HEVC sub tracks.