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

ISO/IEC JTC1/SC29/WG11 MPEG2016/M38648
May 2016, Geneva (CH)

Source	**Telecom ParisTech, Canon Research Centre France**
Status	**For consideration at the 115th MPEG meeting**
Title	**Usage of HEVC Tile Tracks in MPEG-DASH**
Author	Jean Le Feuvre, Cyril Concolato , Franck Denoual, Frédéric Mazé

1 Introduction

This contribution reviews the different tools offered by the HEVC file format for storage of tile tracks and how this can be mapped to MPEG-DASH.

2 HEVC Tile Tracks

HEVC has the ability to divide a frame into rectangular areas called tiles, decodable independently at each frame. Under certain conditions, it is possible to further constrain an HEVC encoder to make each tile independently decodable from the other tiles throughout the lifetime of the video sequence.

2.1 Adaptation using several HEVC Tile Tracks

When constraining the HEVC parameter sets to be identical among different versions of the stream, it is possible to combine the tiles from these versions into a conformant HEVC bitstream, decodable using a single decoder, thereby opening the possibilities to adapt the bitrate on a tile-basis, rather than on a complete sequence level, as shown in Figure 1. This has been demonstrated in [1].

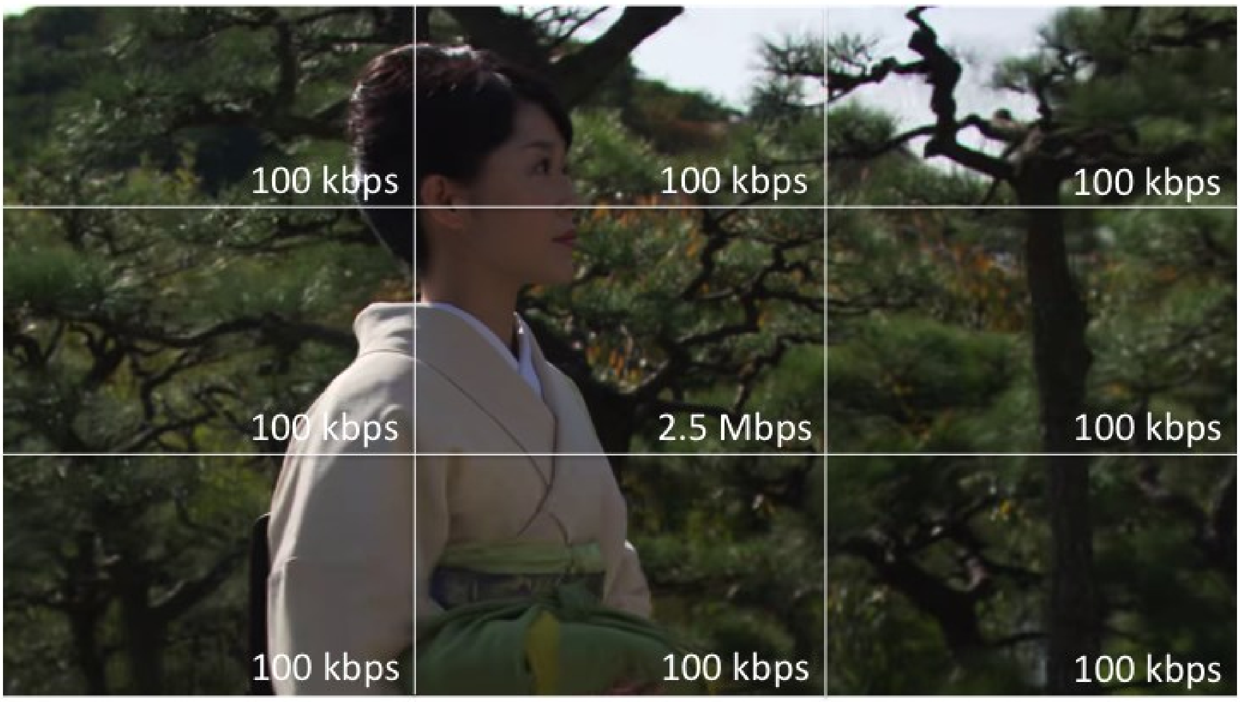

Figure 1 - Tile-based rate adaptation in DASH

Each tile of each quality can typically be packaged in a single track containing only tile-related Video Coding Layer NAL units, and most non Video Coding Layer (non-VCL) NAL units would be in a dedicated track, called “base tile track”.

In such case, reconstruction of the full Access Unit (AU) can be achieved based either on extractors from the base tile track to tile tracks, or on implicit AU reconstruction rules (mostly VCL NALU concatenation rules).

Note: If only a subset of the complete tiles of an HEVC sequence should be decoded then un-needed tile tracks can be discarded and/or some extractors can be ignored while decoding the HEVC sequence; this would however not rebuild a complete image, as shown in Figure 2.

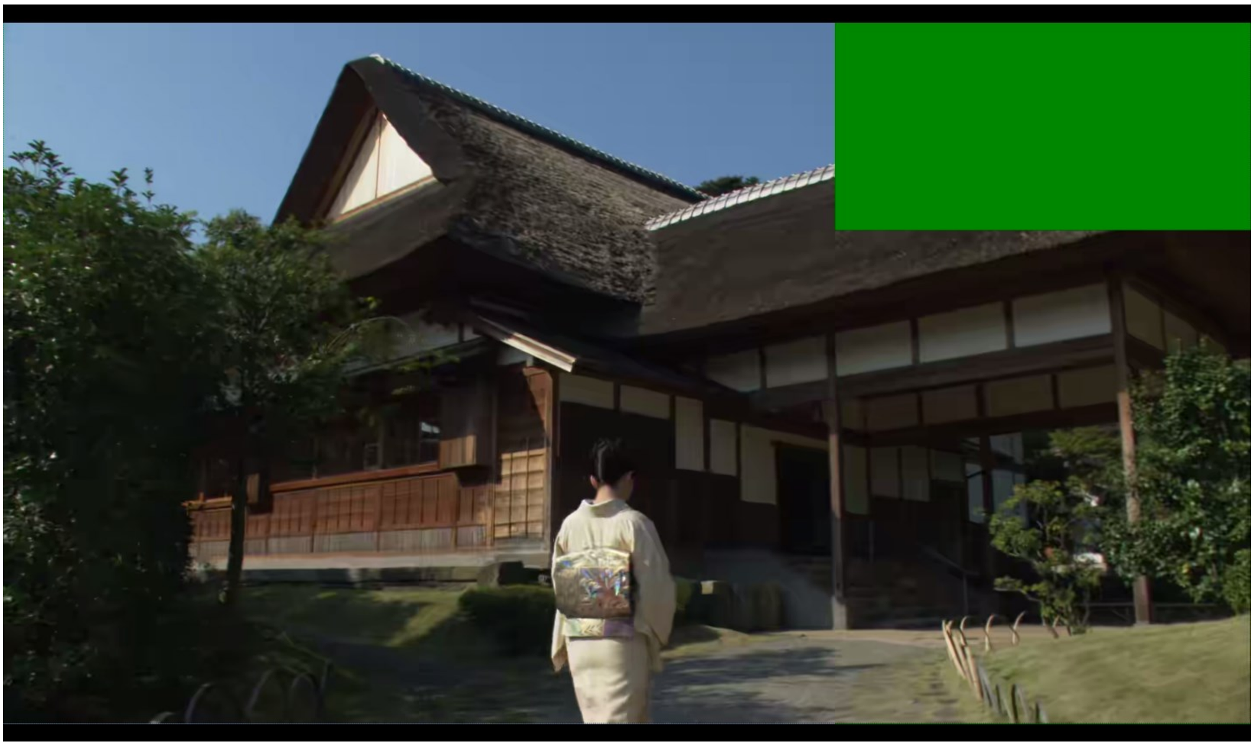

Figure 2 - DASH HEVC with one skipped tile

2.2 Reconstruction of decodable bitstream from HEVC Tile Tracks

The HEVC file format also defines an extractor format giving rules to rewrite part of the bitstream while copying other parts. A typical use case for this is to provide an extractor track that extracts a tile of a NxM motion-constrained tiled HEVC bitstream into a conformant, non-tiled HEVC bitstream with the same resolution as the extracted tile, allowing full-frame playback of a single tile without having to strip part of the reconstructed picture as shown in Figure 3.

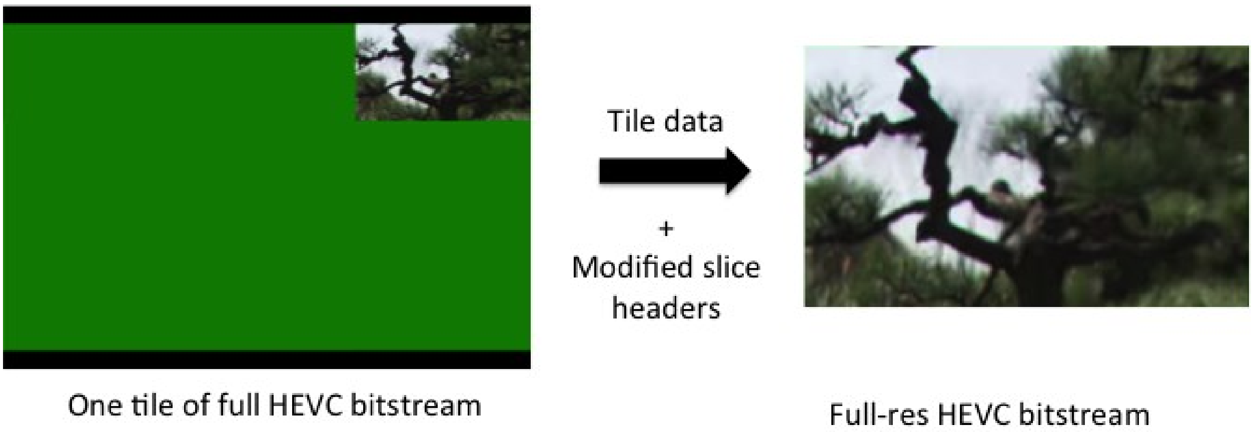

Figure 3 - Rewrite of HEVC tile tracks

Obviously, accessing only the tile of interest through DASH rather than the entire bit stream would save quite a lot of bandwidth. Such a use case may be interesting for ROI inspection, and its usage in DASH should be considered.

3 Mapping of HEVC Tile Tracks to MPEG-DASH: Possible approaches and issues

3.1 Case 1: tile tracks packaged as ‘hvt1’ or ‘lht1’ tracks

In order to perform tile-based access to the video bitstream, the base tile track and tile tracks are each mapped to an MPEG-DASH Representation in its own AdaptationSet, where the tile location is given by an SRD descriptor at the adaptation set level.

Each tile track representation would then have a dependencyId towards the “base tile track”, allowing locating and loading all non-VCL data for that track.

Two approaches would then be possible in order to reconstruct the complete video from the tile tracks.

3.1.1 First approach: tile AdaptationSets only

In this first approach, all tile tracks representations and the base track representation share the same initialization segment, and the DASH client is responsible of fetching in order the different tiles of interest; this simplifies the design of the MPD, but however requires special processing at the DASH client to identify that all tiled representation belong to the same coded object, by analyzing dependency indications, mime type and SRD parameters.

An example MPD for a 2x1 tiling is as follows:

```
<MPD>
<Period >
  <AdaptationSet maxWidth="1280" maxHeight="640" >
    <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
    <SegmentTemplate initialization="v_base.mp4" ... />
    <Representation id="1"
                   mimeType="video/mp4"
                   codecs="hev2.1.6.L186.0"
                   width="1280" height="640"/>
  </AdaptationSet>
  <AdaptationSet maxWidth="640" maxHeight="640" ...>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
    <SegmentTemplate initialization="v_base.mp4" ... />
    <Representation id="1_1"
                   mimeType="video/mp4"
                   codecs="hvt1.1.6.L186.0"
                   dependencyId="1"
                   bandwidth=”128000”/>
    <Representation id="1_2"
                   mimeType="video/mp4"
                   codecs="hvt1.1.6.L186.0"
                   dependencyId="1"
                   bandwidth=”768000”/>
  </AdaptationSet>
  <AdaptationSet maxWidth="640" maxHeight="640" ...>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
    <SegmentTemplate initialization="v_base.mp4" ... />
    <Representation id="2_1"
                   mimeType="video/mp4"
                   codecs="hvt1.1.6.L186.0"
                   dependencyId="1"
                   bandwidth=”128000”/>
    <Representation id="2_2"
                   mimeType="video/mp4"
                   codecs="hvt1.1.6.L186.0"
                   dependencyId="1"
                   bandwidth=”768000”/>
  </AdaptationSet>
 </Period>
</MPD>
```

In this scenario, the DASH client needs to identify that all adaptation sets of type “hvt1” and the same SRD source_id are a single video object, and should not instantiate multiple video decoders. This differs from “regular” logic in DASH (with or without SRD) where each AdaptationSet maps to a single decoder, but is actually very close to multiview use cases (each view in a given adaptation set) or a spatial scalable use case where UHD enhancement and HD base layer would be in separate adaptation set.

3.1.2 Second approach: aggregation AdaptationSet

In this second approach, each tile track representation (or the base track representation) have their own initialization segment, typically including only the tile track and the base track, and an extra “aggregated” adaptation set is used to describe the set of tile tracks composing the full video for each quality; the representations in this set would have their own initialization segment including all tile tracks, and dependencyIds to all tile tracks representation; the media segments of this representation will be empty, as all data is carried in the base track and the tile tracks. This design is a bit heavier but does not require specific processing of the DASH client in order to reconstruct the full video.

An example MPD for a 2x1 tiling is as follows:

```
<MPD>
<Period >
  <AdaptationSet maxWidth="1280" maxHeight="640" >
    <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
    <SegmentTemplate initialization="v_base.mp4" ... />
    <Representation id="1" mimeType="video/mp4" codecs="hev2.1.6.L186.0" width="1280" height="640"/>
  </AdaptationSet>
  <AdaptationSet  ...>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
    <SegmentTemplate initialization="v_tile1.mp4" ... />
    <Representation id="1_1"
                    mimeType="video/mp4"
                    codecs="hvt1.1.6.L186.0"
                    dependencyId="1"
                    bandwidth=”128000”/>
    <Representation id="1_2"
                    mimeType="video/mp4"
                    codecs="hvt1.1.6.L186.0"
                    dependencyId="1"
                    bandwidth=”768000”/>
  </AdaptationSet>
  <AdaptationSet  ...>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
    <SegmentTemplate initialization="v_tile2.mp4" ... />
    <Representation id="2_1"
                    mimeType="video/mp4"
                    codecs="hvt1.1.6.L186.0"
                    dependencyId="1"
                    bandwidth=”128000”/>
    <Representation id="2_2"
                    mimeType="video/mp4"
                    codecs="hvt1.1.6.L186.0"
                    dependencyId="1"
                    bandwidth=”768000”/>
  </AdaptationSet>
  <AdaptationSet  ...>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,1280,640"/>
    <SegmentTemplate initialization="v_all.mp4" ... />
    <Representation id="A" mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_1 2_1"/>
    <Representation id="B" mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_1 2_2"/>
    <Representation id="C" mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_2 2_1"/>
    <Representation id="D" mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_2 2_2"/>
  </AdaptationSet>
 </Period>
</MPD>
```

NOTE: The condition on different initialization segments for the representations comes from the DASH specification on handling of initialization segments with dependent representations. However, since the base tile track cannot be used without the tile tracks, and a single tile track with its base being an incomplete HEVC bitstream, enforcing different initialization segments makes little sense in the tiling case, and this constraint should be removed.

We can notice that this design does not allow expressing adaptation rules of the tile tracks representation, as each aggregated representation explicitly gives the list of dependencies that have to be followed by the DASH engine. One approach to fix this problem would be to list all possible tile combinations in the "aggregated" adaptation set, but this is impractical when using 3x3 or more tiling (for a 2 bitrate 3x3 tiling, that would require 512 combinations!).

3.2 Reconstruction of full HEVC video from HEVC Tile Tracks

In order to perform full HEVC reconstruction from a tile without accessing the entire tiled bitstream, each tile of the video stream would be packaged in a single track of type hvt1, and extraction instruction would be in an additional track of type hev2/hvc2 (since the resulting extracted bitstream is a conformant HEVC bitstream). Both tracks can be packaged in a single file.

An example MPD for a 2x1 tiling is as follows (note that here, both 'hvt1' and 'hev2'/'hvc2' tracks are in the same file):

```
<MPD>
<Period >
  <AdaptationSet maxWidth="1280" maxHeight="640" >
   <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
   <SegmentTemplate initialization="v_base.mp4" ... />
   <Representation    id="1"    mimeType="video/mp4"    codecs="hev2.1.6.L186.0"    width="1280"
height="640"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
   <SegmentTemplate initialization="v_tile1_x.mp4" ... />
   <Representation id="1_1" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
   <Representation id="1_2" mimeType="video/mp4" codecs="hev2.1.6. LXXX.0" dependencyId="1"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
   <SegmentTemplate initialization="v_tile2_x.mp4" ... />
   <Representation id="2_1" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
   <Representation id="2_2" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
  </AdaptationSet>
 </Period>
</MPD>
```

The same problem however arises when trying to have an adaptation set describing the complete video sequence. One would have to explicitly write the combination of at least each tiles of a given quality.

An example MPD for a 2x1 tiling is as follows:

```
<MPD>
<Period >
  <AdaptationSet maxWidth="1280" maxHeight="640" >
   <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
   <SegmentTemplate initialization="v_base.mp4" ... />
   <Representation id="1" mimeType="video/mp4" codecs="hev2.1.6.L186.0" width="1280" height="640"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
   <SegmentTemplate initialization="v_tile1.mp4" ... />
   <Representation id="1_1" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
   <Representation id="1_2" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
   <SegmentTemplate initialization="v_tile2.mp4" ... />
   <Representation id="2_1" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
   <Representation id="2_2" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,1280,640"/>
   <SegmentTemplate initialization="v_all.mp4" ... />
   <Representation mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_1 2_1"/>
   <Representation mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_1 2_2"/>
   <Representation mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_2 2_1"/>
   <Representation mimeType="video/mp4" codecs="hev2.1.6.L186.0" dependencyId="1_2 2_2"/>
  </AdaptationSet>
 </Period>
</MPD>
```

Again, this becomes unreasonable as the number of tiles or qualities increase.

4 Proposed Approach

In order to keep the description of HEVC tiling adaptation in MPD light, we propose the following approach:

- Adaptation Sets containing representations of codec type ‘hvt1’ shall only contain representation of type ‘hvt1’; (i.e. not be mixed with Representations with other codec types)
- Adaptation Sets containing representations of codec type ‘hvt1’ shall contain an SRD descriptor as SupplementalProperty;
- the base tile track of an ‘hvt1’ representation is given by the dependencyId, which shall indicate a representation with codec type hev2/hvc2;
- All ‘hvt1’ representations sharing the same base shall have identical switching and adressing properties as their base tile track: initialization segment, bitstreamSwitching, startWithSAP, segment duration or SegmentTimeline, startNumber, $Time$ or $Number$ addressing;
- the “base tile track” shall be in a dedicated AdaptationSet containing an essential property SRD descriptor, with object_x, object_y, object_width, object_height all being 0. Note that this is not forbidden by Annex H (SRD), it is only undefined.
- Several tile representations, as indicated by the ‘hvt1’ codec type in the MPD, may be gathered in a single adaptation set if and only if they have the same dependencyID and correspond to the same tile, as described by the adaptation set SRD descriptor
- Adaptation Sets containing representations of codec type ‘hvt1’ may be decoded using a single HEVC decoder if and only if:
    - they share the same base tile track, as identified by their dependencyId,
    - they belong to the same SRD group, as identified by the source_id of the SRD descriptor.

This approach avoids defining an “aggregated” adaptation set and works for the above use cases.

An example MPD for a 2x1 tiling combining both use cases could be as follows:

```
<MPD>
<Period >
  <AdaptationSet maxWidth="1280" maxHeight="640" >
   <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,0,0"/>
   <SegmentTemplate initialization="v_base.mp4" ... />
   <Representation id="1" mimeType="video/mp4" codecs="hev1.1.6.L186.0" width="1280" height="640"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
   <SegmentTemplate initialization="v_base.mp4" ... />
   <Representation id="1_1" mimeType="video/mp4" codecs="hvt1.1.6.L186.0" dependencyId="1"/>
   <Representation id="1_2" mimeType="video/mp4" codecs="hvt1.1.6.L186.0" dependencyId="1"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
   <SegmentTemplate initialization="v_base.mp4" ... />
   <Representation id="2_1" mimeType="video/mp4" codecs="hvt1.1.6.L186.0" dependencyId="1"/>
   <Representation id="2_2" mimeType="video/mp4" codecs="hvt1.1.6.L186.0" dependencyId="1"/>
  </AdaptationSet>

  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,0,0,640,640"/>
   <SegmentTemplate initialization="v_tile1_x.mp4" ... />
   <Representation id="1_1_x" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1_1"/>
   <Representation id="1_2_x" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="1_2"/>
  </AdaptationSet>
  <AdaptationSet  ...>
   <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="1,640,0,640,640"/>
   <SegmentTemplate initialization="v_tile2_x.mp4" ... />
   <Representation id="2_1_x" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="2_1"/>
   <Representation id="2_2_x" mimeType="video/mp4" codecs="hev2.1.6.LXXX.0" dependencyId="2_2"/>
  </AdaptationSet>
</Period>
</MPD>
```

In the above example, each tile track is accessible as a single conformant HEVC video through the representations N_K_x, (N being the tile index and K the quality level) while at the same time the complete video can be recomputed by feeding all selected 'hvt1' representations to the HEVC decoder associated with SRD source_id equal to 1.

NOTE: In order to make the proposal less HEVC_centric (ie future-proof), the conditions on 'hvt1' could be replaced by adding an EssentialProperty with schemeID `urn:mpeg:dash:video:tile:2016` for the tile representations and `urn:mpeg:dash:video:basetile:2016` for the "base tile track" representations

5 Conclusion

We have shown major use cases for HEVC tile based adaptation in MPEG-DASH and proposed one approach towards defining how the tiling dependencies can be expressed in a DASH context. We recommend MPEG to use this work as a starting point for HEVC-tiling support in MPEG-DASH to be added as an annex to MPEG-DASH.

6 References

[1] "*Tiled-based Adaptive Streaming using MPEG-DASH* ", J. Le Feuvre and C. Concolato, MMSys, Klagenfurt, Austria, May 2016. More details at https://gpac.wp.mines-telecom.fr/2016/05/25/srd/