

› Proposed design of high-level syntax for spatial relation between independent HEVC sub bitstreams

Submitted by

Alexandre Gabriel (TNO)
Emmanuel Thomas (TNO)

TNO innovation
for life

USE CASES FOR SUB BITSTREAMS

- When only a part of the video must be rendered at one time. Beneficial for reducing decoding computations

Panoramic video with pan and zoom



Aggregation of multiple bitstreams
(e.g. conference call or security camera)

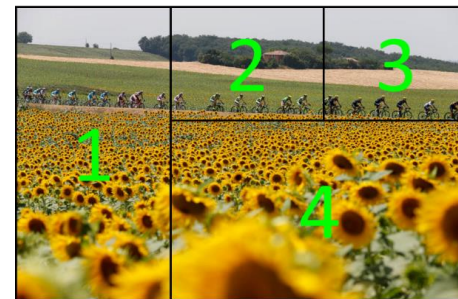


360 video



PROBLEMS WITH CURRENT TILES

- › Tiles were designed for in-frame parallelization - not the use cases presented.
- › Tile size/distribution – tiles must always be in a grid layout which does not allow for the following layouts – one is beneficial for the content (a), the other takes into account the properties of 360 video (b).
- › Tile information is at a low level in the bitstream syntax, making it hard to edit on the fly.



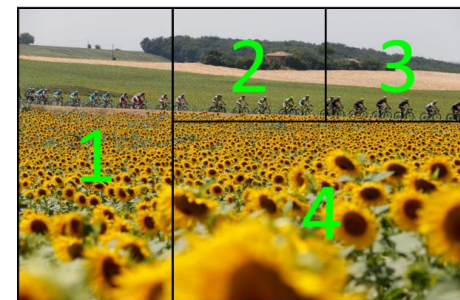
(a)



(b)

PROBLEMS WITH CURRENT TILES

- › Extracting tiles over several frames has well-know challenges.
- › For instance:
 - › In loop filtering must be disabled (since it creates a dependency between the tiles in a frame)
 - › Tile grid needs to be consistent over several frames
 - › The constant tile configuration must be signalled
 - › Motion vectors needs to be constrained within the same tile position over frames
 - › Motion vector tile constraints must be signalled



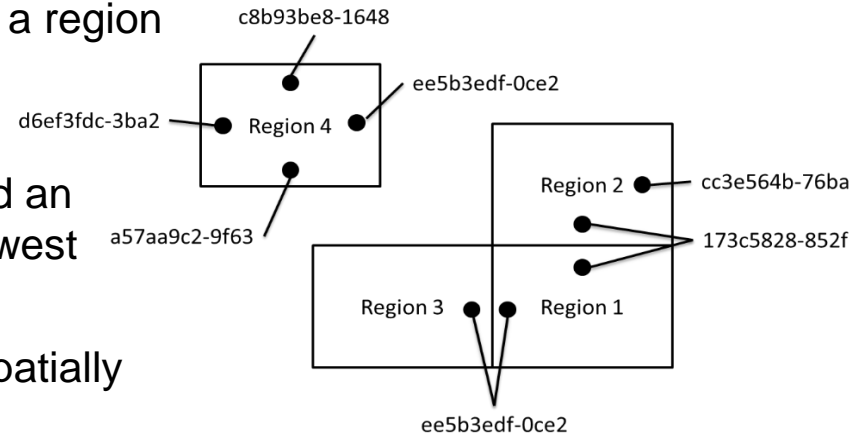
(a)



(b)

PROPOSED DESIGN: HOOKS CONCEPT

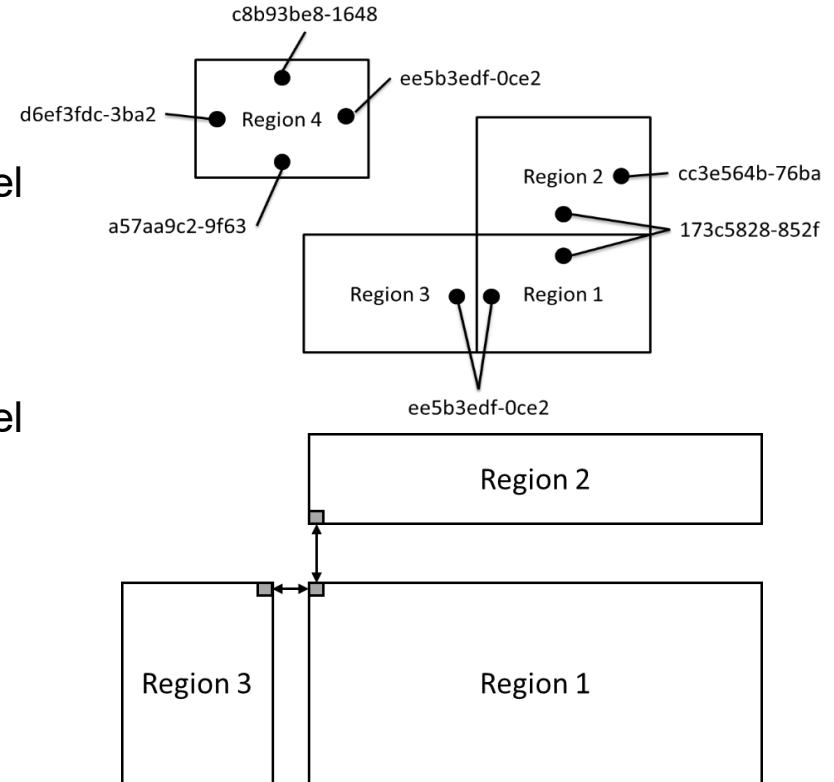
- › Let's call a **tile** an independent sub video bit streams.
- › **DISCLAIMER**: we don't mean by tile the existing HEVC tile concept but a sub bitstream associated with a region in the video.
- › **Hooks** are imaginary points that have an ID and an orientation associated – north, east, south and west
- › When two hooks have matching IDs, tiles are spatially adjacent (e.g. Region 3 and Region 1)



PROPOSED DESIGN: HOOKS CONCEPT

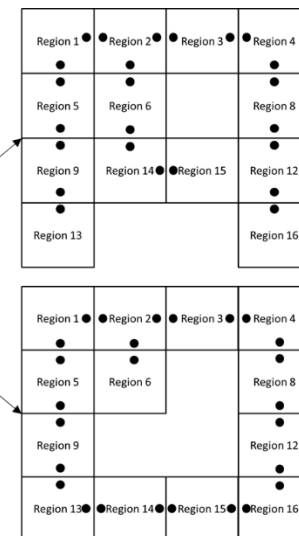
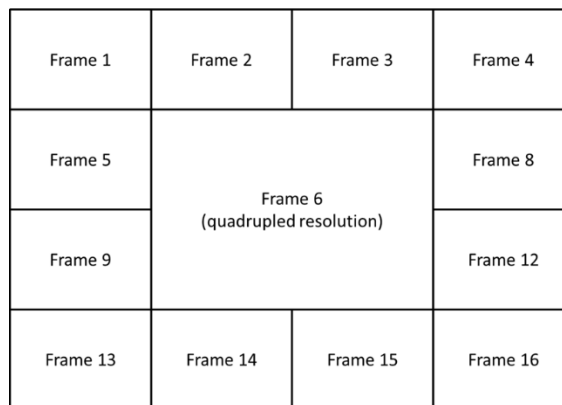
- › Tiles **vertically** adjacent example
 - › The decoder will place region 1's top left pixel below region 2's bottom left pixel

- › Tiles **horizontally** adjacent example
 - › The decoder will place region 1's top left pixel to the right of region 3's top right pixel

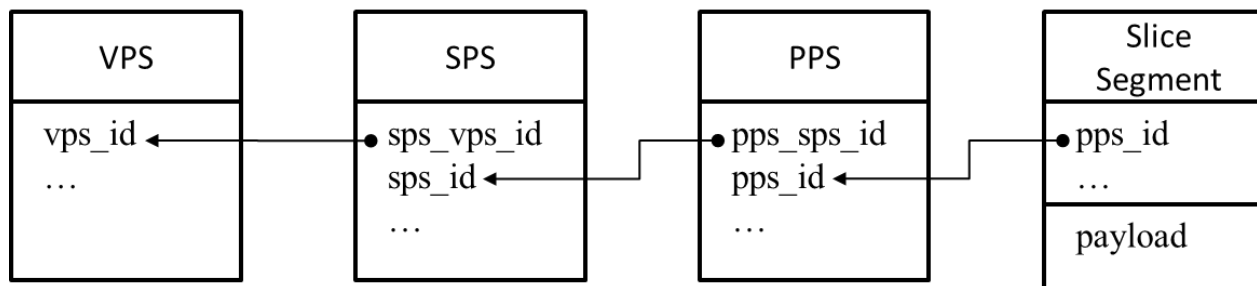


HOOKS CONCEPT

- › Flexible tile layouts can be achieved
- › Tiles with different resolutions can easily coexist
- › The configuration can be deduced by expanding the tiles
- › Different ways of describing the layout are also possible



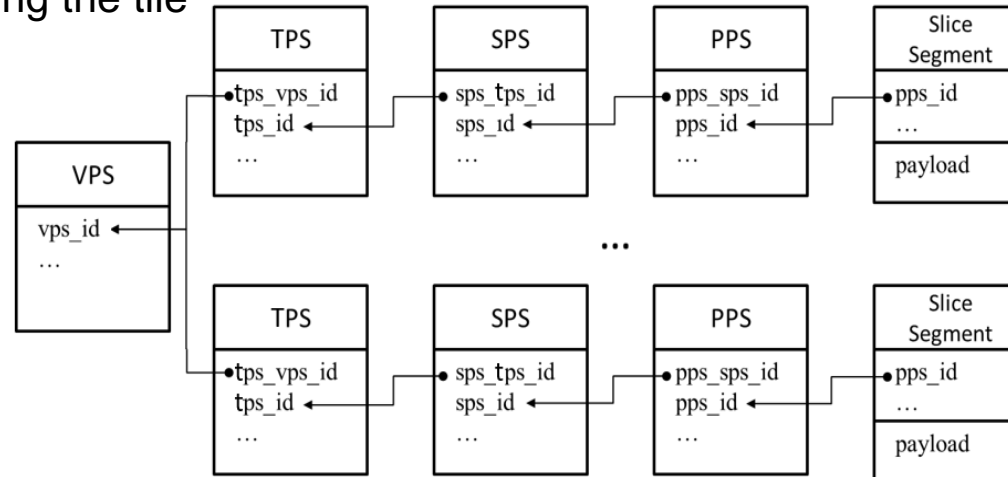
CURRENT HEVC NON-VCL NAL UNIT SYNTAX



- › VPS – Information regarding possible layers in the bitstream
- › SPS – Information about all slices of a sequence; e.g., profile, picture size, etc.
- › PPS – Information that can change from picture to picture, e.g. tile grid, loop filters, etc.

PROPOSED SYNTAX

- › Introduction of new parameter set – Tile-positioning Parameter Set (TPS) which contains information regarding the tile



- › Introduction of new variable in the NAL header – `nuh_tile_id`

PROPOSED SYNTAX

› Addition of Tile-positioning Parameter Set

› tps_video_parameter_set_id

› VPS NAL unit id that the TPS depends on

› tps_tile_positioning_parameter_set_id

› Id of this Tile-positioning Parameter Set

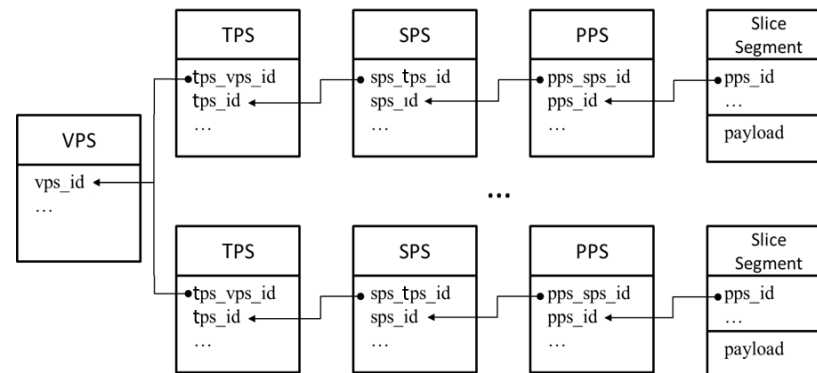
› boundary_identifier_north

› boundary_identifier_east

› boundary_identifier_south

› boundary_identifier_west

› boundary identifier located at the north/east/south/west boundary of the tile



PROPOSED SYNTAX

- › Addition of new parameter in NAL header – `nuh_tile_id`
- › If `nuh_tile_id` $\neq 0$ then it is the `tps_tile_positioning_parameter_set_id` to which the NAL refers to
- › If `nuh_tile_id` = 0 the NAL unit that has no relation with tiles and is invariant to merging and extraction operations of tile bitstreams

<code>nal_unit_header() {</code>	Descriptor
<code>forbidden_zero_bit</code>	<code>f(1)</code>
<code>nal_unit_type</code>	<code>u(6)</code>
<code>nuh_layer_id</code>	<code>u(6)</code>
<code>nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>nuh_tile_id</code>	<code>u(6)</code>
<code>}</code>	

ADVANTAGES

- › Extracting tiles from the bitstream by design (no extra measure)
- › Flexible tile sizes and configurations (homogenous tile size possible)
- › Parallel decoding at sub bitstream level (possibly by 2 decoder instances)
- › NAL parsing of a tile at a high-level in the syntax
 - › Filtering of wanted tiles fast and possible at decoding time
 - › Potentially save on bandwidth usage
 - › Merging 2 two bitstreams (with similar characteristics) into 1 by generating proper non-VCL NAL units relations

CONCLUSION

- › HEVC tiles were not designed for being separately extracted as some use cases required (e.g. 360 viewport-dependent tiled streaming)
- › Hooks allow for flexible and variable tile configuration.
- › Proposed solution introduced 2 new syntax elements:
 - › New parameter set – TPS for describing the relative position of a sub bitstream with one another
 - › New NAL header parameter that specify to which sub bitstream it belong to
- › Recommended to investigate the adoption of the current technical proposal for enabling a true spatial random access by design in HEVC in comparison with ongoing works MCTS, etc...



› **THANK YOU FOR YOUR
ATTENTION**

Take a look:
TIME.TNO.NL

TNO innovation
for life