

JVET-N0496

AHG12: Rectangular tile group address signaling



Rickard Sjöberg, Mitra Damghanian, Martin Pettersson
Ericsson

Background – Rectangular tile groups in VVC



- Rectangular tile groups were adopted at the last JVET meeting
- The position and size of a tile group is signaled using two tile indices, one for the top-left corner and one for the bottom-right corner
- For the first tile group, the top-left tile index is not signaled but inferred to be equal to 0

pic_parameter_set_rbsp() {	Descriptor
...	
single_tile_per_tile_group_flag	u(1)
if(!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {	
num_tile_groups_in_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
top_left_tile_idx[i]	u(v)
bottom_right_tile_idx[i]	u(v)
}	
}	
...	
}	

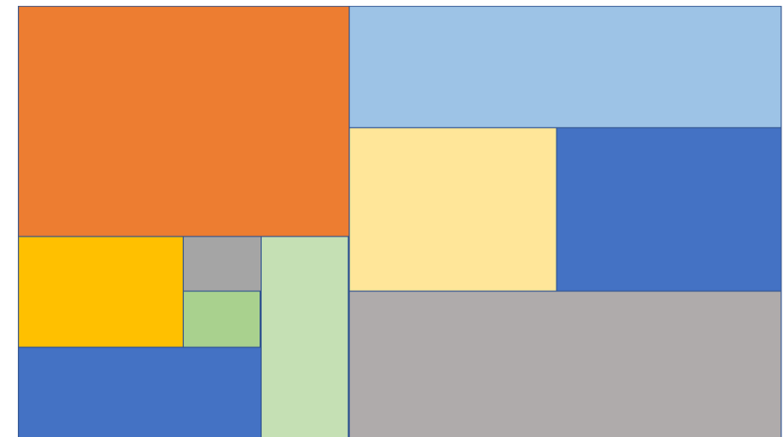
Background – Notes from Marrakesh



JVET-M00853-v2 adds the support of rectangular tile groups in addition to the existing raster scan ordered tile groups, and enables extraction of MCTSs without changing VLC NAL units.

Decision: Adopted, with constraints as follows:

- the tiles in a tile group need to be in raster-scan order.
- the tile groups need to be in increasing address order (see below: dark orange, then light blue, then pale yellow, then dark blue, then light orange, ...)
- The tile group shapes shall be such that each tile, when decoded, shall have its entire left boundary and entire top boundary consisting of a picture boundary or previously decoded tile(s). In other words, the light yellow and light green tiles in the figure below are not allowed.



The following tile group picture was drawn for illustration and discussion purposes regarding the above constraints (an accidental *homage* to Piet Mondrian).

Proposal



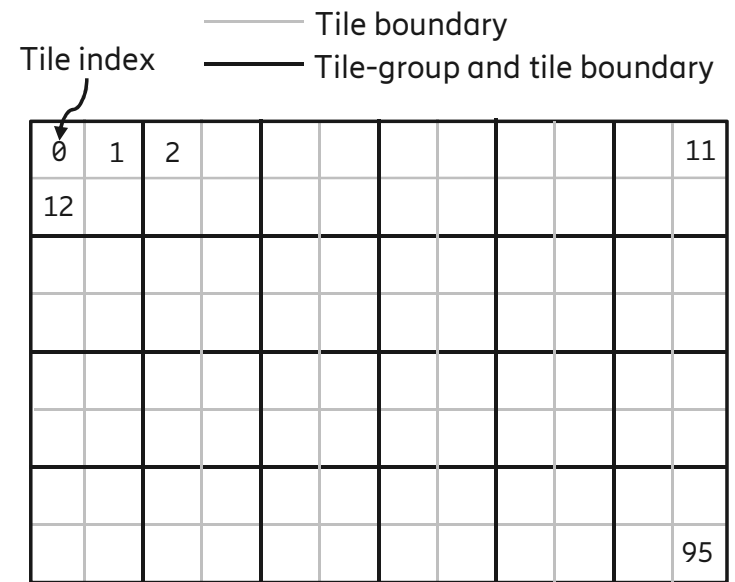
- The signaling of `top_left_tile_idx[ i ]` is redundant
- We propose to remove the `top_left_tile_idx[ i ]` code word and instead derive the value of it as equal to the first tile index in raster scan order that is not yet included in any tile group
- The change would almost cut the number of bits spent on tile indices in half

<code>pic_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code> single_tile_per_tile_group_flag</code>	<code>u(1)</code>
<code> if(!single_tile_per_tile_group_flag)</code>	
<code> rect_tile_group_flag</code>	<code>u(1)</code>
<code> if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {</code>	
<code> num_tile_groups_in_pic_minus1</code>	<code>ue(v)</code>
<code> for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {</code>	
<code> if(i > 0)</code>	<code>-</code>
<code> top_left_tile_idx[i]</code>	<code>u(v)</code>
<code> bottom_right_tile_idx[i]</code>	<code>u(v)</code>
<code> }</code>	
<code> }</code>	
<code>...</code>	
<code>}</code>	

Example



- The CE12 testing conditions specify two test cases with 24 and 96 tile groups
- For a 24 tile group case with 96 tiles the code word lengths would be 7 bits
 - In the VVC draft, that would result in $24 \times 2 \times 7 - 7 = 327$ bits
 - For the proposal, that would result in $24 \times 7 = 168$ bits



Exemplary tile grouping structure

Proposal - semantics



top_left_tile_idx[i] specifies the tile index of the tile located at the top-left corner of the i-th tile group. The value of **top_left_tile_idx[i]** shall not be equal to the value of **top_left_tile_idx[j]** for any i not equal to j. When not present, the value of **top_left_tile_idx[i]** is inferred to be equal to i. The length of the **top_left_tile_idx[i]** syntax element is $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bits.

bottom_right_tile_idx[i] specifies the tile index of the tile located at the bottom-right corner of the i-th tile group. When **single_tile_per_tile_group_flag** is equal to 1 **bottom_right_tile_idx[i]** is inferred to be equal to **top_left_tile_idx[i]**. When not present, the value of **bottom_right_tile_idx[i]** is inferred to be equal to i. The length of the **bottom_right_tile_idx[i]** syntax element is $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bits.

The variable **NumTilesInTileGroup[i]**, which specifies the number of tiles in the i-th tile group, and related variables, are derived as follows:

```
if( i == 0 )
    for( j = 0; j <= num_tiles_minus1; j++ )
        TileInTileGroupFlag[ j ] = 0
for( j = 0; j <= num_tiles_minus1; j++ )
    if( TileInTileGroupFlag[ j ] == 0 )
        break
top_left_tile_idx[ i ] = j
deltaTileIdx = bottom_right_tile_idx[ i ] - top_left_tile_idx[ i ]
NumTileRowsInTileGroupMinus1[ i ] = deltaTileIdx / ( num_tile_columns_minus1 + 1 )
NumTileColumnsInTileGroupMinus1[ i ] = deltaTileIdx % ( num_tile_columns_minus1 + 1 )
NumTilesInTileGroup[ i ] = ( NumTileRowsInTileGroupMinus1[ i ] + 1 ) *
    ( NumTileColumnsInTileGroupMinus1[ i ] + 1 )
for( y = 0; y <= NumTileRowsInTileGroupMinus1[ i ]; y++ )
    for( x = 0; x <= NumTileColumnsInTileGroupMinus1[ i ]; x++ )
        TileInTileGroupFlag[ top_left_tile_idx[ i ] + y * ( num_tile_columns_minus1 + 1 ) + x ] = 1
```

Note - It can be noted that it might be beneficial to replace **NumTileRowsInTileGroupMinus1** and **NumTileColumnsInTileGroupMinus1** by **NumTileRowsInTileGroup** and **NumTileColumnsInTileGroup** and change the derivation of them and the associated semantics in the VVC draft, but that would be purely editorial.

Proposal – option 2 (editorial updates)



- Based on off-line discussions at the Geneva meeting, the following editorially updated semantics is proposed instead
- The new text does not have the break statement that was questioned in one of the Geneva meeting BoG sessions on Coded Picture Regions.

~~top_left_tile_idx[i] specifies the tile index of the tile located at the top-left corner of the i-th tile group. The value of top_left_tile_idx[i] shall not be equal to the value of top_left_tile_idx[j] for any i not equal to j. When not present, the value of top_left_tile_idx[i] is inferred to be equal to i. The length of the top_left_tile_idx[i] syntax element is $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bits.~~

bottom_right_tile_idx[i] specifies the tile index of the tile located at the bottom-right corner of the i-th tile group. ~~When single_tile_per_tile_group_flag is equal to 1 bottom_right_tile_idx[i] is inferred to be equal to top_left_tile_idx[i].~~ When not present, the value of **bottom_right_tile_idx[i]** is inferred to be equal to i. The length of the **bottom_right_tile_idx[i]** syntax element is $\text{Ceil}(\text{Log2}(\text{NumTilesInPic}))$ bits.

The variable **NumTilesInTileGroup[i]**, which specifies the number of tiles in the i-th tile group, and related variables, are derived as follows:

```
if( i == 0 )
    for( j = 0; j <= num_tiles_minus1; j++ )
        TileInTileGroupFlag[ j ] = 0
nextTGIIdx = ( i == 0 ) ? 0 : TopLeftTileIdx[ i-1 ] + NumTileColumnsInTileGroupMinus1[ i-1 ] + 1
while( nextTGIIdx <= num_tiles_minus1 && TileInTileGroupFlag[ nextTGIIdx ] )
    nextTGIIdx++
top_left_tile_idx[ i ] = nextTGIIdx
deltaTileIdx = bottom_right_tile_idx[ i ] - top_left_tile_idx[ i ]
NumTileRowsInTileGroupMinus1[ i ] = deltaTileIdx / ( num_tile_columns_minus1 + 1 )
NumTileColumnsInTileGroupMinus1[ i ] = deltaTileIdx % ( num_tile_columns_minus1 + 1 )
NumTilesInTileGroup[ i ] = ( NumTileRowsInTileGroupMinus1[ i ] + 1 ) *
    ( NumTileColumnsInTileGroupMinus1[ i ] + 1 )
for( y = 0; y <= NumTileRowsInTileGroupMinus1[ i ]; y++ )
    for( x = 0; x <= NumTileColumnsInTileGroupMinus1[ i ]; x++ )
        TileInTileGroupFlag[ top_left_tile_idx[ i ] + y * ( num_tile_columns_minus1 + 1 ) + x ] = 1
```

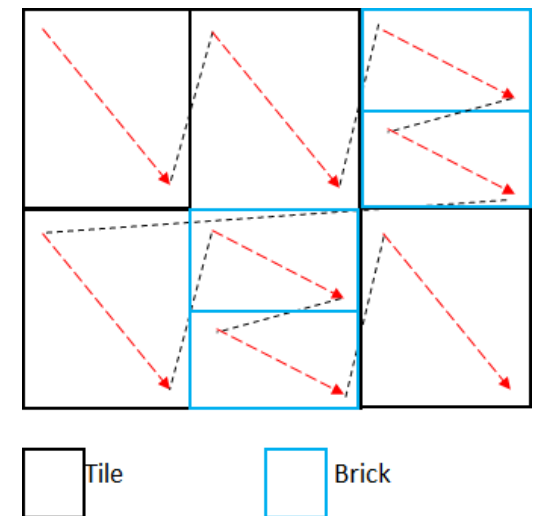
Tiles and bricks



- The v7 of the BoG Report on Coded Picture Regions JVET-N0754-v7 contains the following regarding the proposal on signalling for tile and brick partitioning:

The BoG recommends adoption of N0857 as an implementation of “design in principle” which was agreed in Track B as noted in Main meeting notes.”

- Since the N0857 contribution uses a pre-defined scan order, the syntax element `top_left_tile_idx[i]` can also in N0857 be derived instead of signaled
 - This provides similar savings as in a design not including bricks.
- An implementation of this proposal on top of the spec text provided in JVET-N0857-v1 was provided in an update version of the contribution



Proposal – option 3



pic_parameter_set_rbsp() {	Descriptor
...	
single_brick_per_tile_group_flag	u(1)
if(!single_brick_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag && !single_brick_per_tile_group_flag) {	
num_tile_groups_in_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
top_left_brick_idx[i]	u(v)
bottom_right_brick_idx[i]	u(v)
}	
}	
...	
}	

Proposal – option 3



top_left_brick_idx[i] specifies the brick index of the brick located at the top-left corner of the i-th tile group. The value of **top_left_brick_idx[i]** shall not be equal to the value of **top_left_brick_idx[j]** for any i not equal to j. When not present, the value of **top_left_brick_idx[i]** is inferred to be equal to i. The length of the **top_left_brick_idx[i]** syntax element is $\text{Ceil}(\text{Log2}(\text{NumBricksInPic}))$ bits.

bottom_right_brick_idx[i] specifies the brick index of the brick located at the bottom-right corner of the i-th tile group. When **single_brick_per_tile_group_flag** is equal to 1, **bottom_right_brick_idx[i]** is inferred to be equal to **top_left_brick_idx[i]**. When not present, the value of **bottom_right_brick_idx[i]** is inferred to be equal to i. The length of the **bottom_right_brick_idx[i]** syntax element is $\text{Ceil}(\text{Log2}(\text{NumBricksInPic}))$ bits.

It is a requirement of bitstream conformance that any particular tile shall only be included in one tile group.

The variable **TopLeftBrickIdx[i]**, **NumBricksInTileGroup[i]** and **BricksToTileGroupMap[j]**, which specifies the brick index of the first brick in the tile group, the number of bricks in the i-th tile group and the mapping of bricks to tile groups, are derived as follows:

```
if( i == 0 )
    for( j = 0; j < NumBricksInPic; j++ )
        BrickInTileGroupFlag[ j ] = 0
nextTGIIdx = 0
while( nextTGIIdx < NumBricksInPic && BrickInTileGroupFlag[ nextTGIIdx ] )
    nextTGIIdx++
TopLeftBrickIdx[ i ] = nextTGIIdx
NumBricksInTileGroup[ i ] = 0
for( j = 0; j < NumBricksInPic; j++ ) {
    if ( BrickColBd[ j ] >= BrickColBd[ top_left_brick_idx TopLeftBrickIdx [ i ] ] &&
        BrickColBd[ j ] <= BrickColBd[ bottom_right_brick_idx[ i ] ] &&
        BrickRowBd[ j ] >= BrickRowBd[ top_left_brick_idx TopLeftBrickIdx [ i ] ] &&
        BrickRowBd[ j ] <= BrickColBd[ bottom_right_brick_idx[ i ] ] ) {
        NumBricksInTileGroup[ i ]++
        BricksToTileGroupMap[ j ] = i
        BrickInTileGroupFlag[ j ] = 1
    }
}
```

