INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 04 MPEG VIDEO CODING

ISO/IEC JTC 1/SC 29/WG 04 m63684

July 2023, Geneva, Switzerland

Title **[MIV] On indexing of syntax elements in the extended geometry assistance SEI**
Source **PUT**
Authors **Adrian Dziembowski, Dawid Mieloch, Błażej Szydełko**

Abstract

This document proposes to change indexing of syntax elements within the EGA SEI, addressing issue #499: https://mpeg.expert/software/MPEG/Video/MIV/Specifications/23090-12/-/issues/499

1 Syntax and semantics

Change of variable names:

- sbl -> i
- sbc -> j
- lvl -> k

Other possible name changes / changes to be discussed:

- recursive_split_function -> recursive_split_structure (?)
- rsf_ -> rss (?)

F.2.8.3 Block-based geometry features: recursive split function

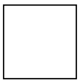
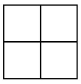
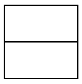
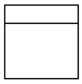
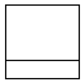
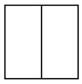
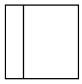
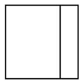
recursive_split_function(v, i, j, k) {	**Descriptor**
if(k < bbgf_max_number_of_splits[v]) {	
rsf_split_flag[v][i][j][k]	u(1)
}	
if(k < bbgf_max_number_of_splits[v] && rsf_split_flag[v][i][j][k]) {	
rsf_quad_split_flag[v][i][j][k]	u(1)
if(rsf_quad_split_flag[v][i][j][k]) {	
recursive_split_function(v, i, j, k + 1)	
recursive_split_function(v, i, j + 1, k + 1)	
recursive_split_function(v, i + 1, j, k + 1)	
recursive_split_function(v, i + 1, j + 1, k + 1)	
} else {	
rsf_split_orientation_flag[v][i][j][k]	u(1)

rsf_split_symmetry_flag[v][i][j][k]	u(1)
if(!rsf_split_symmetry_flag[v][i][j][k]) {	
rsf_split_first_block_bigger[v][i][j][k]	u(1)
}	
if(rsf_split_orientation_flag[v][i][j][k]) {	
recursive_split_function(v, i, j, k + 1)	
recursive_split_function(v, i, j + 1, k + 1)	
} else {	
recursive_split_function(v, i, j, k + 1)	
recursive_split_function(v, i + 1, j, k + 1)	
}	
}	
} else {	
rsf_skip_flag[v][i][j][k]	u(1)
if(!rsf_skip_flag[v][i][j][k]) {	
if(i == 0 && j == 0) { /* none */	
LTMinFlag = 2	
LTMaxFlag = 2	
} else if(i == 0) { /* left */	
LTMinFlag = 0	
LTMaxFlag = 0	
} else if(j == 0) { /* top */	
LTMinFlag = 1	
LTMaxFlag = 1	
} else {	
rsf_ltmin_flag[v][i][j][k]	u(1)
rsf_ltmax_flag[v][i][j][k]	u(1)
LTMinFlag = rsf_ltmin_flag[v][i][j][k]	
LTMaxFlag = rsf_ltmax_flag[v][i][j][k]	
}	
rsf_zmin_delta[v][i][j][k]	se(v)
rsf_zmax_delta[v][i][j][k]	se(v)
}	
}	
}	

F.3.9.3 Block-based geometry features: recursive split function

Table F-1 indicates the available split types (including no split) of a block, with the associated values of the block division syntax. Figure F-1 shows an example of the block subdivision of a view.

Table F-1 — The different block split types and associated values of the syntax elements

								
rsf_split_flag [v][i][j][k]	0	1	1	1	1	1	1	1
rsf_quad_split_flag [v][i][j][k]	-	1	0	0	0	0	0	0
rsf_split_orientation_flag [v][i][j][k]	-	-	0	0	0	1	1	1
rsf_split_symmetry_flag [v][i][j][k]	-	-	1	0	0	1	0	0
rsf_split_first_block_bigger [v][i][j][k]	-	-	-	0	1	-	0	1

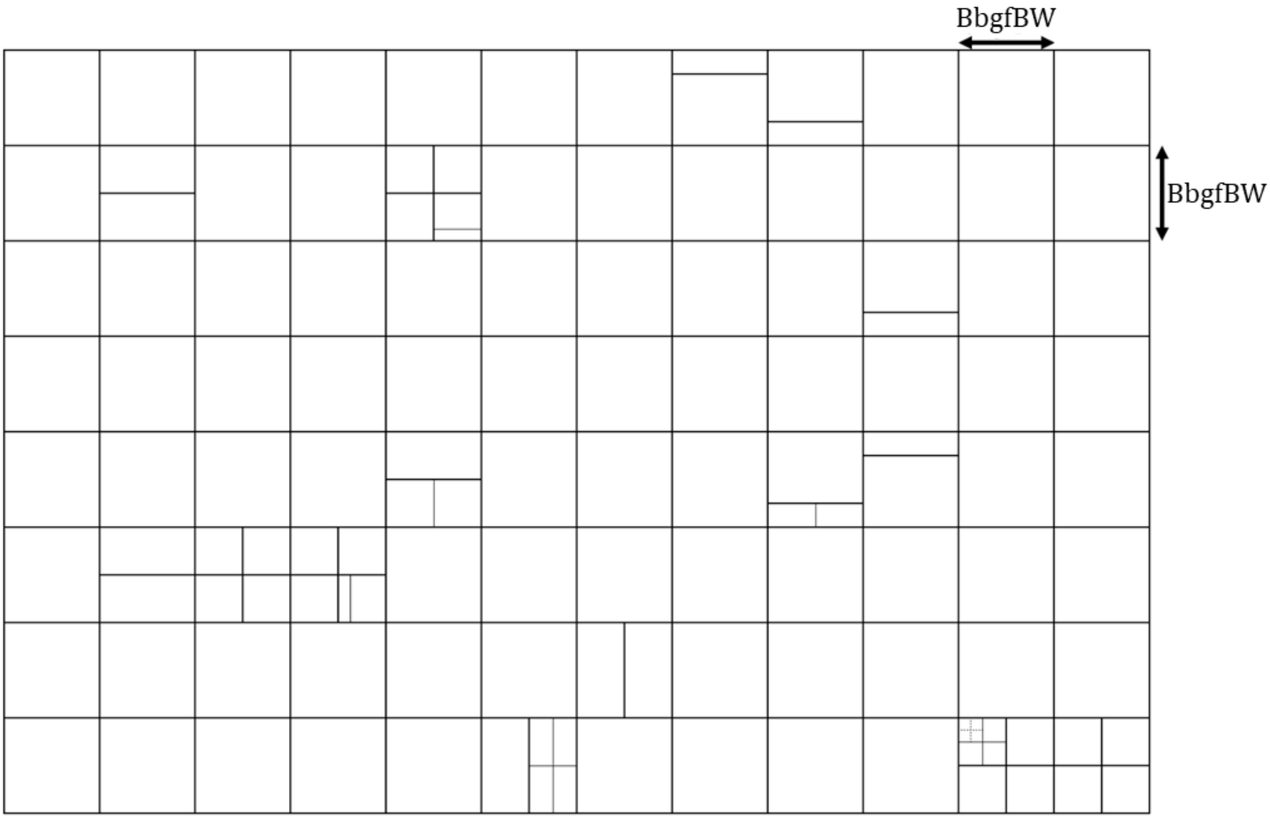


Figure F-1 — Example of a partition of a view into possible block divisions with bbgf_max_number_of_splits[v] set to 3

rsf_split_flag[v][i][j][k] equal to 1 indicates that the ~~current~~ block with indexes i, j, k of the view with index v is split into smaller subblocks. rsf_split_flag[v][i][j][k] equal to 0 indicates that the ~~current~~ block with indexes i, j, k of the view with index v is not split into smaller subblocks.

rsf_quad_split_flag[v][i][j][k] equal to 0 indicates that the ~~current~~ block with indexes i, j, k of the view with index v is split into two rectangular subblocks. rsf_quad_split_flag[v][i][j][k] equal to 1 indicates that the current block with indexes i, j, k of the view with index v is split into four square subblocks of identical sizes.

rsf_split_orientation_flag[v][i][j][k] equal to 0 indicates that the ~~current~~ block with indexes i, j, k of the view with index v is split horizontally. rsf_split_orientation_flag[v][i][j][k] equal to 1 indicates that the ~~current~~ block with indexes i, j, k of the view with index v is split vertically.

rsf_split_symmetry_flag[v][i][j][k] equal to 0 indicates that the area of the two sublocks of the block with indexes i, j, k of the view with index v differ, with the division occuring at a quarter of the block width from one end. rsf_split_symmetry_flag[v][i][j][k] equal to 1 indicates that the area of the two sublocks of the block with indexes i, j, k of the view with index v is equal.

rsf_split_first_block_bigger[v][i][j][k] equal to 1 indicates that the first subblock of the block with indexes i, j, k of the view with index v (top subblock if rsf_split_orientation_flag[v][i][j][k] is equal to 0, and left subblock if rsf_split_orientation_flag[v][i][j][k] is equal to 1) is bigger than the second subblock. rsf_split_first_block_bigger[v][i][j][k] equal to 0 indicates that the first subblock of the block with indexes i, j, k of the view with index v is smaller than the second subblock.

rsf_skip_flag[v][i][j][k] equal to 0 indicates that rsf_zmin_delta[v][i][j][k] and rsf_zmax_delta[v][i][j][k] syntax elements are present in the bitstream, and that rsf_ltmin_flag[v][i][j][k] and rsf_ltmax_flag[v][i][j][k] may be present. rsf_skip_flag equal[v][i][j][k] to 1 indicates that no other syntax elements are present in the bitstream for the ~~current~~ block with indexes i, j, k of the view with index v, and it suggests that the geometry information in this block has not changed since the previous frame in display order.

rsf_ltmin_flag[v][i][j][k] equal to 0 indicates that the prediction of the ~~current~~ minimum geometry of the block with indexes i, j, k of the view with index v is to be taken from the left block, otherwise from the top block.

rsf_ltmax_flag[v][i][j][k] equal to 0 indicates that the prediction of the ~~current~~ maximum geometry of the block with indexes i, j, k of the view with index v is to be taken from the left block, otherwise from the top block.

rsf_zmin_delta[v][i][j][k] specifies the remainder to be added to the prediction to obtain the minimum geometry value suggested for the ~~current~~ block with indexes i, j, k of the view with index v.

rsf_zmax_delta[v][i][j][k] specifies the remainder to be added to the prediction to obtain the maximum geometry value suggested for the ~~current~~ block with indexes i, j, k of the view with index v.

Variables *ZMinLeft* and *ZMaxLeft* are set to the minimum and maximum geometry range of the left block, respectively, and if available.

Variables *ZMinTop* and *ZMaxTop* are set to the minimum and maximum geometry range of the top block, respectively, and if available.

The suggested minimum geometry range *ZMin* and maximum geometry range *ZMax* of the ~~current~~ block with indexes i, j, k of the view with index v are derived by the following formulae:

$$ZMin = (LTMinFlag == 2 ~?~ 0 : LTMinFlag == 1 ~?~ ZMinTop : ZMinLeft + \\ + \mathrm{bbgf\_qs}[v] * \mathrm{rsf\_zmin\_delta}[v][i][j][k] \tag{F-1}$$

$$ZMax = (LTMaxFlag == 2 ~?~ 0 : LTMaxFlag == 1 ~?~ ZminTop : ZminLeft) + \\ + \mathrm{bbgf\_qs}[v] * \mathrm{rsf\_zmax\_delta}[v][i][j][k] \tag{F-2}$$

2 Recommendation

We recommend adopting the proposed syntax changes into the Working Draft.

3 Acknowledgement

The research was supported by the Ministry of Science and Higher Education of Republic of Poland.