

MEDIATEK

Interaction between dual tree and minimum CU size

Tzu-Der Chuang, Ching-Yeh Chen, Chih-Wei Hsu,
Yu-Wen Huang, Shaw-Min Lei

Presented by Tzu-Der (Peter) Chuang
16th Meeting: Geneva, CH,
1–11 October 2019

Overall Summary

- In VVC Draft 6, when dual tree is enabled and MinCbSizeY is set to 128, a CTU will be partitioned to four 64x64 blocks with inferred quad-tree split, which is contradictory to the MinCbSizeY setting
- Propose 3 methods to solve this issue
 1. qtbtt_dual_tree_intra_flag is inferred to be 0 in the SPS when MinCbSizeY is 128
 2. qtbtt_dual_tree_intra_flag is signalled, but the dual tree is not applied when MinCbSizeY is 128
 3. When qtbtt_dual_tree_intra_flag is 1, MinCbSizeY shall not be larger than 64

Problem Definition

- In VVC Draft 6, a variable of MinCbSizeY is derived to indicate the minimum CU size of this sequence
 - $\text{MinCbSizeY} = 1 \ll (\log2_min_luma_coding_block_size_minus2 + 2)$
 - The $\log2_min_luma_coding_block_size_minus2$ is signaled in the SPS
- However, if the MinCbSizeY is set to 128 and the dual tree coding structure is enabled (e.g. `qtbtt_dual_tree_intra_flag` is true), a 128x128 CTU will be inferred divided into 64x64 blocks in I-slice
- This is contradictory to the MinCbSizeY setting

Proposed Method-1

- In VVC Draft 6, both the `qtbtt_dual_tree_intra_flag` and `log2_min_luma_coding_block_size_minus2` are signalled in SPS
 - The `log2_min_luma_coding_block_size_minus2` is signaled after the `qtbtt_dual_tree_intra_flag`
- Propose to
 - Move the `qtbtt_dual_tree_intra_flag` syntax after the `log2_min_luma_coding_block_size_minus2`
 - `qtbtt_dual_tree_intra_flag` is inferred to be 0 in the SPS when `MinCbSizeY` is 128

seq_parameter_set_rbsp() {	Descriptor
...	
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
...	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_ctu_size_minus5	u(2)
log2_min_luma_coding_block_size_minus2	ue(v)
if(ChromaArrayType != 0 && log2_min_luma_coding_block_size_minus2 < 5)	
qtbtt_dual_tree_intra_flag	u(1)
...	
}	

Proposed Method-2

- qtbt_dual_tree_intra_flag is signalled, but the dual tree is not applied when MinCbSizeY is 128
 - The single tree coding structure is applied

coding_tree_unit() {	Descriptor
...	
if(slice_type == I && qtbt_dual_tree_intra_flag && MinCbSizeY <= 64)	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbSizeY, 0)	
else	
coding_tree(xCtb, yCtb, CtbSizeY, CtbSizeY, 1, 1, 0, 0, 0, 0, 0, SINGLE_TREE, MODE_TYPE_ALL)	
}	

Proposed Method-3

- When dual tree is enabled, MinCbSizeY shall not be larger than 64
- An encoder normative constraint is added for `log2_min_luma_coding_block_size_minus2`

`log2_min_luma_coding_block_size_minus2` plus 2 specifies the minimum luma coding block size. It is a requirement of bitstream conformance that when the `qtbtt_dual_tree_intra_flag` is 1, the value of `log2_min_luma_coding_block_size_minus2` shall not be larger than or equal to 5.

Conclusions

- This contribution presents three methods to satisfy the minimum CU size constraint with considering the dual tree
 1. `qtbtt_dual_tree_intra_flag` is inferred to be 0 in the SPS when `MinCbSizeY` is 128
 2. `qtbtt_dual_tree_intra_flag` is signalled, but the dual tree is not applied when `MinCbSizeY` is 128
 3. When `qtbtt_dual_tree_intra_flag` is 1, `MinCbSizeY` shall not be larger than 64



everyday genius