



JVET-N0190

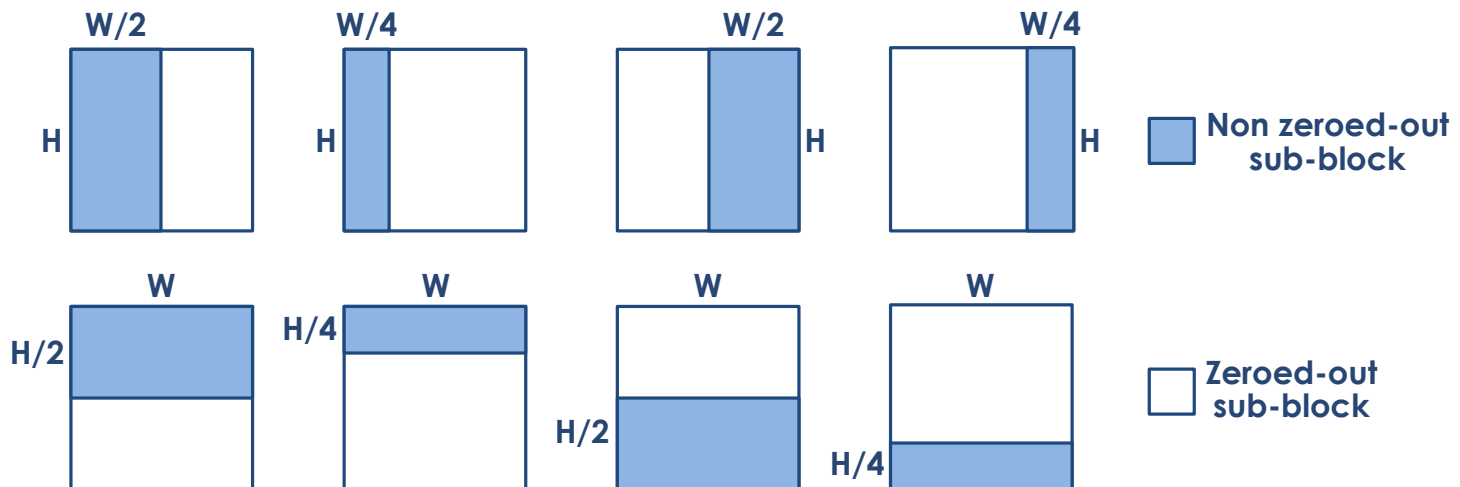
Non-CE6: Modifications on sub-block transform

J. Jung, D. Kim, G. Ko, J. Son, J. Kwak (WILUS)

March, 2019

○ Introduction

- Sub-block transform codes inter-predicted residues using sub-TU tiling.
 - Residues of one sub-block are coded using implicit transform type combinations.
 - `tu_cbf_luma` is signaled since it can be 0 or 1.
 - Residues of the remaining sub-block are zeroed out.
 - `tu_cbf_luma` is inferred to be equal to 0.



○ Proposed method

- Sub-block transform is not allowed when `tu_cbf_luma` of non zeroed-out sub-block is 0.
 - `tu_cbf_luma` of each sub-block is not parsed and it is inferred according to sub-TU index and position of the residual TU.

transform unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	Descriptor
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex == 0 && cu_sbt_pos_flag) ((subTuIndex == 1 && !cu_sbt_pos_flag)))) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfLuma)))	
tu_cbf_luma [x0][y0]	ae(v)
InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[x0][y0]	
}	
...	

subTuIndex	cu_sbt_pos_flag	tu_cbf_luma	
		VTM 4	Proposed method
0	0	signaled	Inferred (1)
0	1	Inferred (0)	Inferred (0)
1	0	Inferred (0)	Inferred (0)
1	1	signaled	Inferred (1)



Experimental results

- VTM-4.0 under the CTC

	Random access Main10				
	Over VTM-4.0				
	Y	U	V	EncT	DecT
Class A1	0.01%	0.04%	0.02%	100%	100%
Class A2	-0.01%	-0.02%	0.08%	100%	100%
Class B	0.01%	-0.07%	-0.04%	100%	99%
Class C	-0.01%	-0.02%	0.05%	100%	101%
Class E					
Overall	0.00%	-0.02%	0.02%	100%	100%
Class D	-0.02%	-0.15%	-0.03%	100%	101%
Class F	-0.01%	0.08%	0.06%	100%	100%

	Low delay B Main10				
	Over VTM-4.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	-0.02%	-0.12%	0.19%	100%	99%
Class C	-0.07%	0.40%	0.20%	100%	99%
Class E	-0.02%	0.03%	-0.03%	100%	99%
Overall	-0.03%	0.09%	0.14%	100%	99%
Class D	-0.04%	-0.15%	-0.67%	100%	98%
Class F	-0.17%	0.73%	0.13%	100%	100%



○ Conclusion

- In this contribution
 - It is proposed to not allow SBT when `tu_cbf_luma` of a part of the partitioned TUs equal to 0.
 - Therefore, `tu_cbf_luma` is not parsed and inferred using the residual position flag.
 - It is shown that the proposed method does not degrade the coding efficiency.
 - RA: 0.00% (Y) / 100% (EncT) / 100% (DecT)
 - LD: -0.03% (Y) / 100% (EncT) / 99% (DecT)
 - The proposed method makes the parsing condition check for `tu_cbf_luma` much simpler than that of the current spec.
- Thanks to Tencent for cross-check

