

JVET-Q0186

CE3-related: TB-level residual coding selection for lossless coding

Zhi-Yi Lin, Tzu-Der Chuang, Ching-Yeh Chen, Chih-Wei Hsu, Yu-Wen Huang,
Shaw-Min Lei (MediaTek)

A. Nalci, H. Wang, M. Coban, M. Karczewicz (Qualcomm)

Overall Summary

- Propose block-level switch between regular residual coding (RRC) and transform skip residual coding (TSRC) for better coding efficiency and flexibility without adding any new low-level syntax
- Method 1: Use transform skip (TS) flag to select between RRC and TSRC
 - A slice-level flag is signalled to specify whether transform is disabled or not
- Method 2: Further allows TS flag to select between RRC and TSRC for BDPCM blocks
 - Another slice-level flag is signalled to specify whether TS flag is inferred as 0 for BDPCM coded blocks

		AI			RA			LB		
		Bit savings	EncT	DecT	Bit savings	EncT	DecT	Bit savings	EncT	DecT
Method 1	CTC	-5.46%	114	108	-5.73%	113	102	-5.06%	106	106
	TGM	-3.88%	-	-	-4.52%	-	-	-4.53%	-	-
Method 2	CTC	-7.86%	136	94	-6.33%	117	100	-5.91%	114	105
	TGM	-8.50%	-	-	-6.37%	-	-	-5.45%	-	-

Introduction

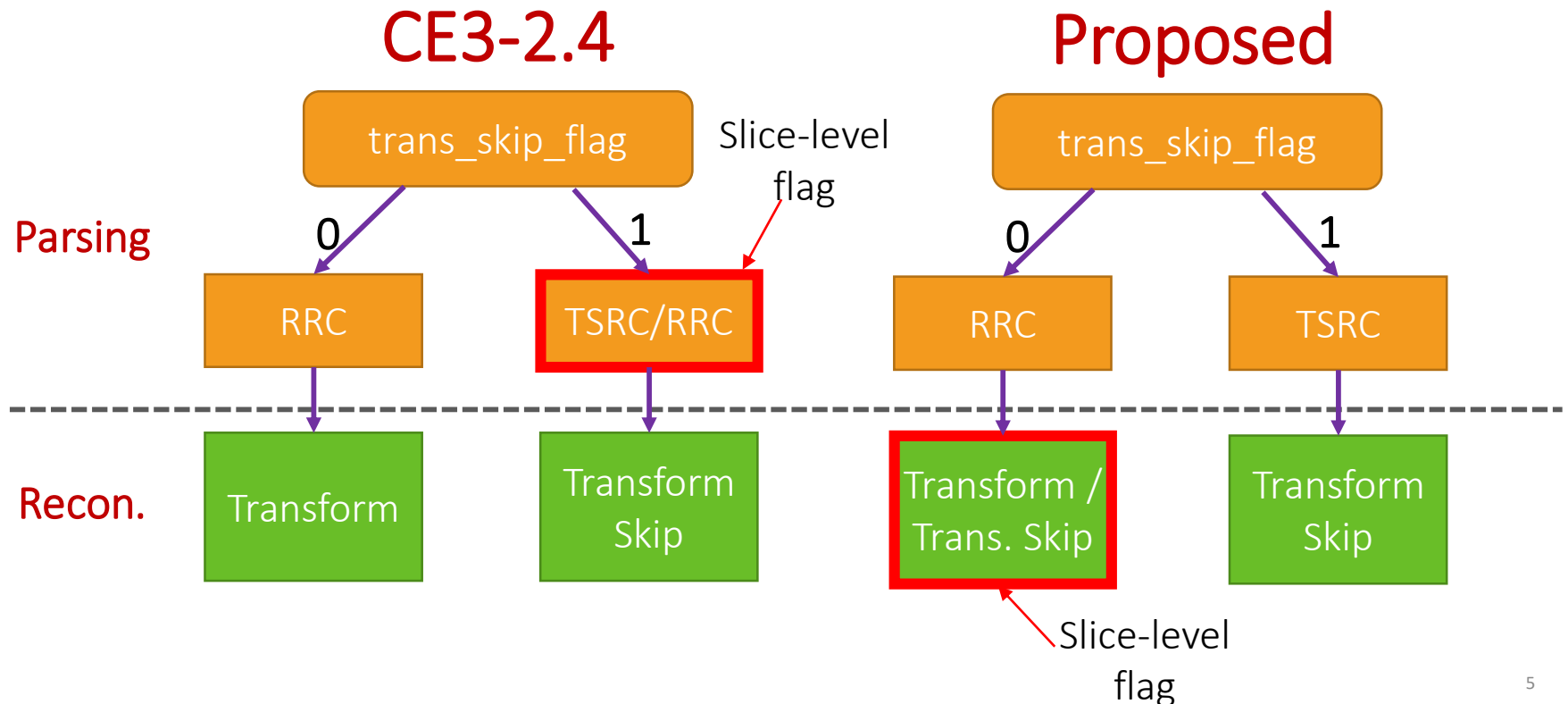
- In VTM7.0, lossless coding is achieved by forcing encoder to choose TS mode, so TSRC is always applied
- It was observed that RRC brings more bit savings for lossless coding for some sequences
- Several CE3 proposals proposed to use high level syntax to select between RRC or TSRC for TS mode for the **entire slice**
 - Lack of flexibility
- Propose two methods to support block-level switch between RRC and TSRC without adding any new low-level syntax

Proposed Method 1 – Disable Transform

- A high-level syntax to disable the transform process in decoding process
 - A PPS flag and a slice-level flag are added
- When transform is disabled:
 - The transform process and the quantization process are the same as those when TS mode is applied
 - The TS flag will be used to select the residual method between RRC and TSRC
 - Achieve block-level switching by using the TS flag

Comparison – Proposed Method vs. CE3-2.4

- Different place to switch functionality
 - CE3-2.4: switch RRC/TSRC in parsing stage
 - Proposed: switch transform or transform skip in reconstruction stage



Results - Method 1 (Anchor: VTM7.0 Lossless)

	All Intra			Random Access			Low delay B		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM7	Test		VTM7	Test		VTM7	Test	
Class A1	2.2	2.4	-5.67%	2.2	2.4	-6.47%			
Class A2	1.6	1.7	-8.99%	1.7	1.9	-8.25%			
Class B	2.2	2.3	-4.22%	2.3	2.4	-3.95%	2.3	2.4	-3.95%
Class C	1.9	2.0	-5.20%	2.4	2.6	-5.52%	2.4	2.6	-5.43%
Class D	1.9	2.0	-4.90%	2.8	2.9	-5.31%	2.7	2.9	-5.27%
Class E	2.8	2.9	-4.10%				3.1	3.3	-6.43%
Class F	5.3	5.5	-3.59%	33.7	35.0	-4.34%	50.7	52.3	-4.20%
TGM	11.8	12.3	-3.88%	107.1	112.4	-4.52%	124.9	131.3	-4.53%
Overall with Classes D, F, TGM excluded	2.1	2.3	-5.46%	2.2	2.4	-5.73%	2.6	2.7	-5.06%
Enc Time[%]	114%			113%			106%		
Dec Time[%]	108%			102%			106%		

Thank Kwai for cross-checking!!

Proposed Method 2 – Disable TS Inference for BDPCM Blocks

- A high-level syntax to disable the inference of TS flag for BDPCM coded blocks
 - A PPS flag and a slice-level flag are added
- When disabling the inference of always using TS for BDPCM blocks, the TS flag is parsed for BDPCM blocks
 - The TS flag will be used to select the residual method between RRC and TSRC
 - Achieve block-level switching for BDPCM mode

Proposed Method 2 – Disable TS Inference for BDPCM Blocks

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Descriptor
...	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA) {	
if(sps_transform_skip_enabled_flag && (!BdpcmFlag[x0][y0][0] slice_bdpcm_infer_transform_skip_disabled_flag) && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && !cu_sbt_flag)	
transform_skip_flag[x0][y0][0]	ae(v)
if(!transform_skip_flag[x0][y0][0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	

Results - Method 2 (BDPCM: 2) (Anchor: VTM7.0 Lossless)

	All Intra			Random Access			Low delay B		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM7	Test		VTM7	Test		VTM7	Test	
Class A1	2.2	2.4	-6.81%	2.2	2.4	-7.21%			
Class A2	1.6	1.8	-10.56%	1.7	1.9	-8.59%			
Class B	2.2	2.3	-6.42%	2.3	2.5	-4.63%	2.3	2.4	-4.64%
Class C	1.9	2.1	-7.30%	2.4	2.6	-6.10%	2.4	2.6	-6.00%
Class D	1.9	2.1	-9.06%	2.8	2.9	-6.09%	2.7	2.9	-5.86%
Class E	2.8	3.1	-9.35%				3.1	3.4	-7.93%
Class F	5.3	5.9	-8.66%	33.7	36.2	-6.53%	50.7	53.4	-5.85%
TGM	11.8	13.0	-8.50%	107.1	114.7	-6.37%	124.9	132.1	-5.45%
Overall with Classes D, F, TGM excluded	2.1	2.3	-7.86%	2.2	2.4	-6.33%	2.6	2.7	-5.91%
Enc Time[%]	136%			117%			114%		
Dec Time[%]	94%			100%			105%		

Thank Kwai for cross-checking!!

Conclusion

- Propose block-level switch between RRC and TSRC for better coding efficiency and flexibility without adding any new low-level syntax
- Method 1: Disable transform at high-level
- Method 2: Disable TS inference for BDPCM blocks

		AI			RA			LB		
		Bit savings	EncT	DecT	Bit savings	EncT	DecT	Bit savings	EncT	DecT
Method 1	CTC	-5.46%	114	108	-5.73%	113	102	-5.06%	106	106
	TGM	-3.88%	-	-	-4.52%	-	-	-4.53%	-	-
Method 2	CTC	-7.86%	136	94	-6.33%	117	100	-5.91%	114	105
	TGM	-8.50%	-	-	-6.37%	-	-	-5.45%	-	-

Thank you!

Results - Method 2 (BDPCM: 2)

(Anchor: VTM7.0 Lossless + BDPCM: 2)

	All Intra			Random Access			Low delay B		
	ratio		bit-rate savings	ratio		bit-rate savings	ratio		bit-rate savings
	VTM7	Test		VTM7	Test		VTM7	Test	
Class A1	2.3	2.4	-5.55%	2.3	2.4	-6.29%			
Class A2	1.6	1.8	-8.77%	1.7	1.9	-8.14%			
Class B	2.2	2.3	-3.84%	2.4	2.5	-3.84%	2.4	2.4	-3.84%
Class C	2.0	2.1	-4.77%	2.4	2.6	-5.41%	2.4	2.6	-5.31%
Class D	2.0	2.1	-4.55%	2.8	2.9	-5.27%	2.8	2.9	-5.24%
Class E	2.9	3.1	-4.52%				3.2	3.4	-6.32%
Class F	5.7	5.9	-3.40%	34.9	36.2	-4.38%	51.9	53.4	-4.06%
TGM	12.5	13.0	-3.70%	109.4	114.7	-4.43%	127.2	132.1	-3.87%
Overall with Classes D, F, TGM excluded	2.2	2.3	-5.27%	2.2	2.4	-5.61%	2.6	2.7	-4.95%
Enc Time[%]	136%			117%			114%		
Dec Time[%]	94%			100%			105%		