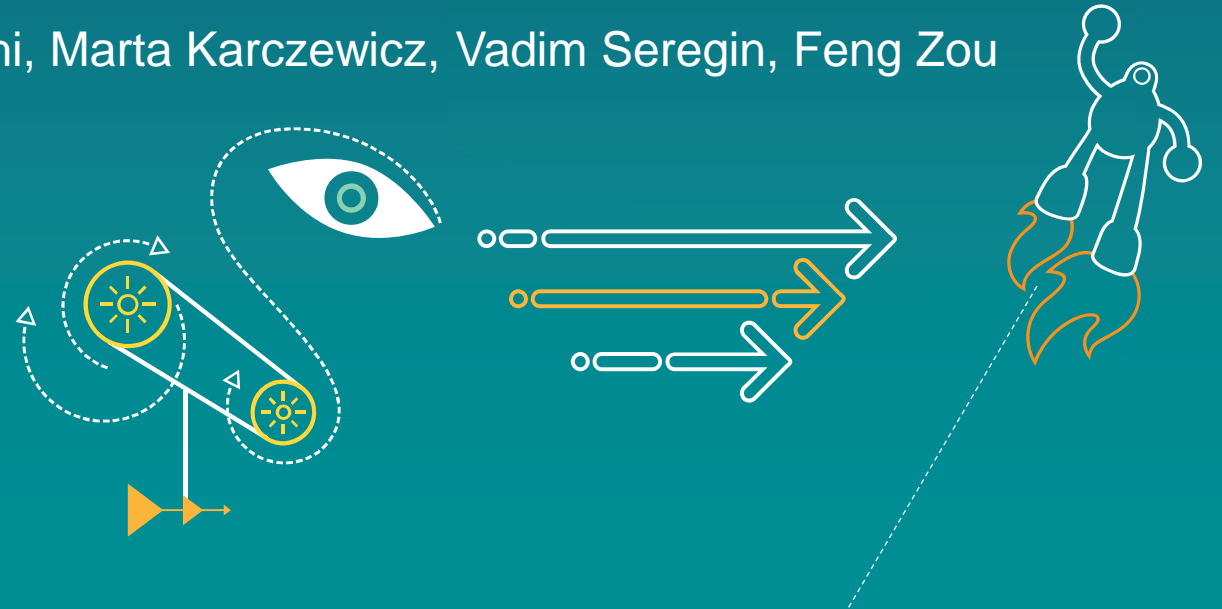# Palette Mode Semantics, Codeword, and Encoder Fixes

Qualcomm®

Wei Pu, Rajan Joshi, Marta Karczewicz, Vadim Seregin, Feng Zou

Semantic Fixes: Palette Size and Palette Predictor List Size

- Related syntax elements

     In SPS header, palette_max_size and palette_max_predictor_size

- Issue: they are unbounded

     Codeword of **palette_predictor_run** and **palette_num_signalled_entries** can be longer than 32 bits

- Semantic fixes

     palette_max_size must be no larger than 64x64 (largest CU)

     palette_max_predictor_size must be no larger than 2x64x64 (twice largest CU)

Codeword Fix

- ## In SCM3.0
  - **palette_num_signalled_entries** uses Truncated Unary (TU) code
  - It can grow over 32 bits

- ## Propose to replace TU with other code on the table
  - Candidate 1. HEVC coefficient coding *xWriteCoefRemainExGolomb()*
  - Candidate 2. EG0 *xWriteEpExGolomb()*
  - Candidate 3. Palette run length coding *xWriteTruncMsbP1RefinementBits()*
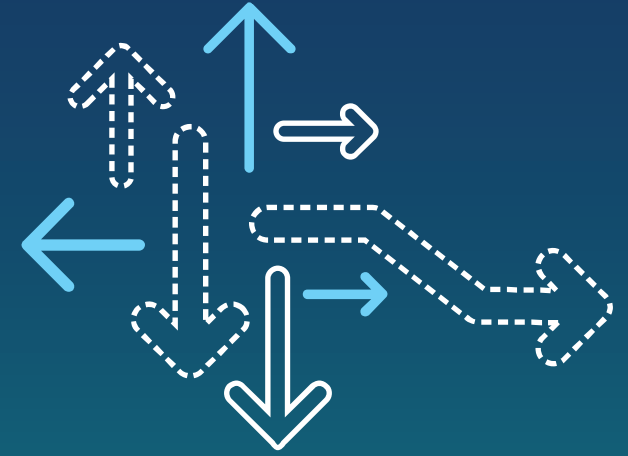
Results

Results v.s. SCM3.0 Anchor

	HEVC Coeff. code			EG0			PLT run code		
	All Intra			**All Intra**			**All Intra**		
	G/Y	B/U	R/V	G/Y	B/U	R/V	G/Y	B/U	R/V
RGB, text & graphics with motion, 1080p & 720p	-0.01%	-0.01%	-0.01%	-0.02%	-0.02%	-0.02%	-0.06%	-0.03%	-0.05%
RGB, mixed content, 1440p & 1080p	-0.01%	-0.01%	-0.01%	-0.01%	0.00%	0.00%	-0.01%	-0.02%	-0.02%
RGB, Animation, 720p	0.00%	0.00%	0.00%	0.00%	0.01%	0.02%	0.00%	0.00%	0.00%
RGB, camera captured, 1080p	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-0.03%	0.00%	0.02%
YUV, text & graphics with motion, 1080p & 720p	0.01%	-0.01%	0.00%	-0.02%	0.01%	0.01%	-0.01%	0.04%	0.01%
YUV, mixed content, 1440p & 1080p	0.00%	-0.01%	-0.01%	0.00%	0.00%	0.00%	-0.01%	0.07%	0.08%
YUV, Animation, 720p	0.00%	0.00%	0.00%	0.01%	0.04%	0.04%	0.02%	0.04%	0.02%
YUV, camera captured, 1080p	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

- Conclusion
  1. All candidates show similar results;
  2. Suggest to use EG0 as it minimizes editorial work.

	…	…	…
palette_coding( )	palette_num_signalled_entries	~~TR~~, EG0	~~cMax~~ = 31, cRiceParam = 0, -
	…	…	…

Alternative Approach I

In JCTVC-S1005 SCC Draft Spec:

palette_num_signalled_entries	TR	cMax = 31, cRiceParam = 0

Actually, it should be:

palette_num_signalled_entries	TR	cMax = (palette_max_size − paletteNumPredictedEntries 31), cRiceParam = 0

To avoid codeword over 32:

palette_num_signalled_entries	TR	cMax = min(32, palette_max_size − paletteNumPredictedEntries 31), cRiceParam = 0

Alternative Approach II, simplest!!!

- Semantically restrict palette_num_signaled_entries ≤ 31
- Done!

Encoder Only Fix

In SCM3.0

Palette encoder generates unused indices in non 'palette share' mode

Reasons

1. RD check in *derivePLTLossy()*
2. *reorderPLT()* may generate some unused entries

   Ref: JCTVC-T0063 for detail explanation

Suggest

Fix this issue to improve reference software quality

Encoder Fix v.s. SCM3.0 Anchor

	All Intra		
	G/Y	B/U	R/V
RGB, text & graphics with motion, 1080p & 720p	-0.1%	-0.1%	-0.1%
RGB, mixed content, 1440p & 1080p	0.0%	-0.1%	-0.1%
RGB, Animation, 720p	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	0.0%	-0.1%	-0.1%
YUV, mixed content, 1440p & 1080p	0.0%	-0.1%	-0.1%
YUV, Animation, 720p	0.0%	0.0%	0.0%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%

Thank you!
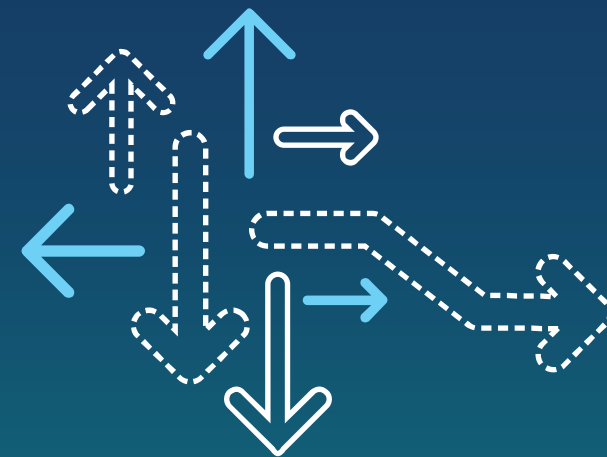
Encoder Only Fix (Detail)

In SCM3.0

Palette encoder generates unused indices in non 'palette share' mode

Reasons

1. RD check in *derivePLTLossy()*

   Ref: JCTVC-T0063 for detail explanation

2. *reorderPLT()* may generate some unused entries

   E.g, palette = [(0, 0, 0), (10, 10, 10)]

   Pixel block = [(5, 5, 5), (10, 10, 10), (10, 10, 10), (10, 10, 10)]

   ➔indices are {0, 1, 1, 1}.

   After calling reorderPLT(), palette = [(10, 10, 10), (0, 0, 0)]

   ➔indices are {0, 0, 0, 0}. ➔ Index 1 is signaled but not used

Suggest

Fix this issue to improve reference software quality