

REDEFINING MOBILITY



# AhG5 and AhG18: Bypass of the significance and coefficient level flags for higher throughput

## JCTVC-O0209

Rajan Joshi, Joel Sole, Marta Karczewicz

# Introduction

- Extension of the concept proposed in O0208 to significance information
- **Proposal:** Bypass the significance and coefficient level flags for 4×4 sub-blocks.
- Two different options for application:
  - Method 1: The Rice parameter for the previous 4×4 subblock is above a threshold
  - Method 2: The initial Rice parameter as proposed in RCE2 D1 (O0239) is above a threshold.
- Reduction in the number of coded bins in the average and in the worst-case

# Method 1

- In HEVC, for each  $4 \times 4$  sub-block, the Rice parameter is initialized to 0 and updated for each coefficient
- Use the Rice parameter at the end of the previous sub-block to bypass the coefficient level flags
  - If `previousRiceGolomb`  $\geq 4$ , then *coeff\_abs\_level\_remaining* encodes the absolute level of the coefficient (minus 1 if it is the last significant coefficient in the TU).
  - *coeff\_abs\_level\_remaining* is coded before sign.

# Method 1 - continued

HEVC	Proposed Method
Encode <i>coded_sub_block_flag</i> Loop to encode <i>significance_coeff_flag</i> Loop to encode <i>coeff_abs_level_greater1_flag</i> Loop to encode <i>coeff_abs_level_greater2_flag</i> Loop to encode <i>coeff_sign_flag</i> Loop to encode <i>coeff_abs_level_remaining</i>	<b>If ( previousRiceGolomb &lt; 4 )</b> Encode <i>coded_sub_block_flag</i> Loop to encode <i>significance_coeff_flag</i> Loop to encode <i>coeff_abs_level_greater1_flag</i> Loop to encode <i>coeff_abs_level_greater2_flag</i> Loop to encode <i>coeff_sign_flag</i> Loop to encode <i>coeff_abs_level_remaining</i> <b>Otherwise</b> Loop to encode <i>coeff_abs_level_remaining</i> Loop to encode <i>coeff_sign_flag</i>

# Method 2

- In RCE2 test D1, it is proposed to
  - Initialize the Rice parameter to a value different from 0 depending on statistics
  - Increase the maximum Rice parameter (which currently is 4)
- On top of RCE2 test D1, method 1 is modified in two ways:
  1. Threshold on Rice parameter larger than 4
  2. Re-use of the initial Rice parameter derived by method RCE2 test D1 (instead of storing the parameter of the previous sub-block)
- Replace **if (previousRiceGolomb < 4)** by **if (initialRiceGolomb < 5)**
  - HEVC v1 behavior is achieved whenever initial Rice parameter is less than or equal to 4
  - Thresholds other than 5 are also tested.

# Results

- BD-rates for AhG18 test conditions for high bit-depth (HBD)
- Method 2 results provided on top of RCE2 test D1 results

# Performance for HBD

**Method 1**

	All Intra		
	Y	U	V
12-bit	1.2%	1.3%	1.3%
14-bit	0.5%	0.6%	0.6%
16-bit	0.0%	0.0%	0.0%
Enc Time[%]	89%		
Dec Time[%]	96%		

**Method 2**

	All Intra		
	Y	U	V
12-bit	-7.2%	-6.8%	-6.8%
14-bit	-18.8%	-18.5%	-18.4%
16-bit	-28.7%	-28.5%	-28.6%
Enc Time[%]	92%		
Dec Time[%]	89%		

**Threshold = 5**

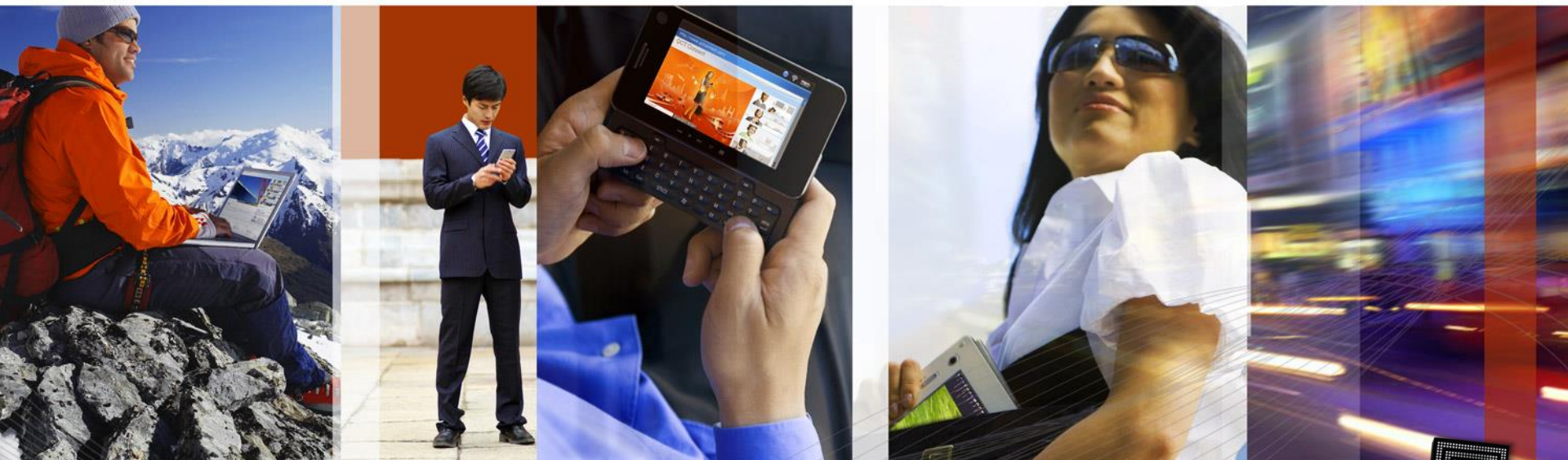
**Method 2  
(anchor RCE2 test D1)**

	All Intra		
	Y	U	V
12-bit	0.0%	0.0%	0.0%
14-bit	-0.1%	-0.1%	-0.1%
16-bit	-0.1%	-0.1%	-0.1%
Enc Time[%]	106%		
Dec Time[%]	105%		

# Conclusions

- Method to increase throughput for high bit-depths by bypassing coefficient flags
  - Only *coeff\_abs\_level\_remaining* and signs are coded
- No performance impact for HBD





REDEFINING MOBILITY



# AhG5 and AhG18: Bypass of the significance and coefficient level flags for higher throughput

## JCTVC-O0209

Rajan Joshi, Joel Sole, Marta Karczewicz

# Bin usage example

CrowdRun 16 bit QP = -37

Total	Regular	Bypass
60.52	6.20	54.32
58.20	1.76	56.44

ParkJoy 16 bit QP = -37

Total	Regular	Bypass
58.2	6.20	51.69
55.5	1.76	53.75