

JVET-0618

CE2-RELATED: REMOVAL OF LUMA- CHROMA DEPENDENCY IN LMCS

Zhipin Deng
ByteDance



JVET-0618: Removal of luma-chroma dependency in LMCS

■ Problem statement

- *Current CRS cannot be started until all the luma prediction samples of the whole block are available → potential luma-chroma latency*

■ Proposal

- *A fixed chroma scaling factor derivation to remove the latency*
 - The chroma scaling factor is derived once for a I slice and used for all blocks that refer to the same LMCS model
 - CRS factor is identified from a representative piece-wise function index,
 - *Method #1: directly use the median bin index*
 - $(\text{lmcs_min_bin_idx} + \text{LmcsMaxBinIdx}) \gg 1$
 - *Method #2: indexed from a representative luma value in reshaped domain*
 - $(\text{ForwardLUT}(0) + \text{ForwardLUT}(1 \ll \text{BitDepthY}) - 1) \gg 1$

Simulation results (SDR)

■ Apply the proposed CRS to single tree only (CTC)

		PSNR-Y	PSNR-U	PSNR-V
Method #1	RA	-0.01%	-0.03%	-0.03%
	LDB	0.01%	0.06%	-0.11%
Method #2	RA	-0.01%	-0.08%	-0.07%
	LDB	0.01%	0.09%	-0.14%

■ Apply the proposed CRS to both single tree and dual tree (non-CTC)

		PSNR-Y	PSNR-U	PSNR-V
Method #1	AI	-0.01%	-0.06%	-0.13%
	RA	0.02%	-0.70%	-0.73%
	LDB	0.05%	-0.68%	-0.76%
Method #2	AI	-0.01%	-0.05%	-0.14%
	RA	0.02%	-0.77%	-0.75%
	LDB	0.04%	-0.69%	-0.78%

Simulation results (HDR)

- Apply the proposed CRS to both single tree and dual tree
 - *when the reference is HDR with LMCS turned ON (non-CTC)*

LQP0 LMCS1 plus the Proposed vs. HDR LQP0 LMCS1				
		wPSNR-Y	wPSNR-U	wPSNR-V
Method #1	AI	0.00%	0.00%	0.00%
	RA	-0.01%	-0.47%	-1.11%
Method #2	AI	1.46%	-13.77%	-17.80%
	RA	1.13%	-15.52%	-19.20%

- *when the reference is HDR CTC anchor (CTC)*

LQP0 LMCS1 plus the Proposed vs. HDR CTC anchor (LQP1 LMCS0)				
		wPSNR-Y	wPSNR-U	wPSNR-V
Method #1	AI	-1.64%	14.15%	8.34%
	RA	-1.61%	13.87%	5.89%
Method #2	AI	-0.20%	-2.68%	-12.07%
	RA	-0.49%	-4.14%	-13.73%

Conclusions

- Benefits
 - *Totally remove the dependency between luma and chroma in LMCS*
 - *A fixed chroma scaling factor is calculated only once per slice*
 - *No coding performance loss*
- It is recommended adopting one of the proposed methods

Thanks LGE for crosschecking!

THANK YOU!