

CE8 Subset2: A Joint Proposal on Improving the Adaptive Loop Filter in TMuC0.9 by MediaTek, Qualcomm, and Toshiba

Ching-Yeh Chen, Chih-Ming Fu, Chia-Yang Tsai, Yu-Wen Huang, Shawmin Lei, Marta Karczewicz, In Suk Chong, Tomoo Yamakage, Takeshi Chujoh, and Takashi Watanabe



Presented by Yu-Wen Huang
4th JCT-VC Meeting in Daegu
20-28 January, 2011

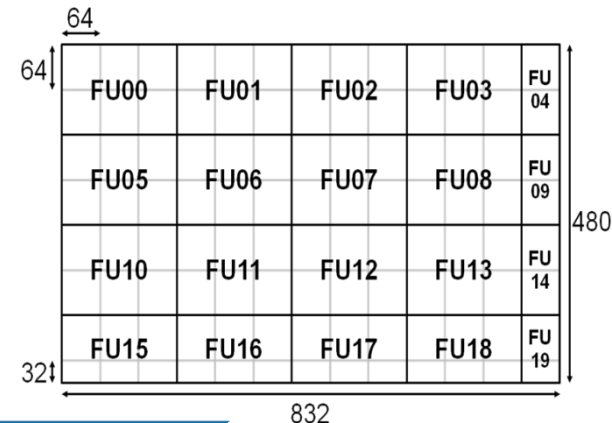
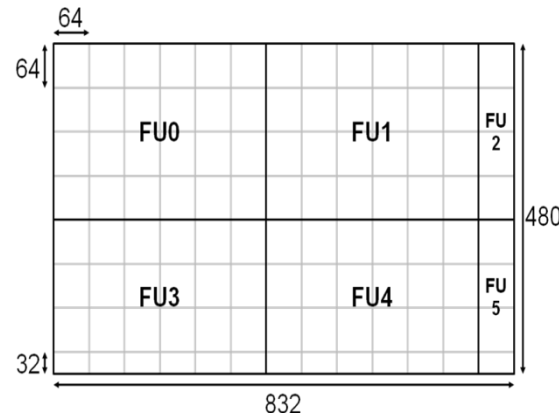
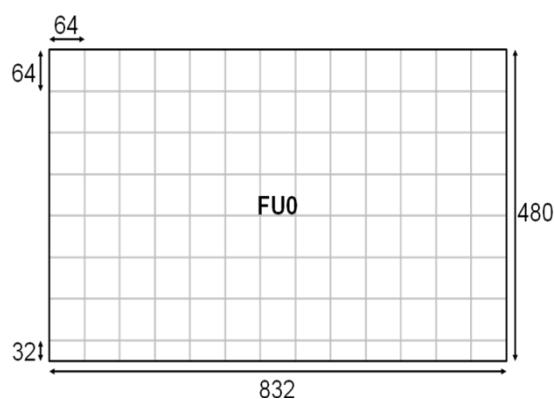
Overall Summary

- MediaTek, Qualcomm, and Toshiba's ALF (MQT_ALF)
 - Filter Unit (FU) based local adaptation
 - Pixel-Adaptation (PA) method, used in the current HM
 - Add two low complexity methods
 - Single Filter (SF), Adaptive Offset (AO)
 - Reduce number of encoding passes from 16 to 1 (encoder-only)
 - Significantly less external memory access and power consumption
- Better coding efficiency and less decoding time

	HE-AI	HE-RA	HE-LD
BD-Rate	0.1%	-0.4%	-1.9%
Encoding Time Reduction	0%	2%	2%
Decoding Time Reduction	15%	15%	17%

Filter Unit for Local Adaptation

- Each picture can be divided into equal-size FUs
 - Except for image boundaries
- FU size can be signaled in the slice header
- Each FU can use one of the following methods
 - Pixel Adaptation (PA)
 - Single Filter (SF)
 - Adaptive Offset (AO)



Pixel-Adaptation (PA) Method

- Similar to the ALF in HM1.0
- Classify pixels into groups according to sum of Laplacian with a reduced window size (from 7x7 to 3x3)
- One filter for each group
- Filter can be 3x3 square, 5x5 square, 7x7 diamond, or 9x9 diamond.

Single-Filter (SF) Method

- A degenerative case of the PA method
 - When the final number of groups is 1
 - Used by the entire FU, not pixel by pixel
 - Simply skip pixel classification operations
- The filter can be new or time-delayed
 - New: coefficients sent
 - Time-delayed: not any coefficient sent

Adaptive-Offset (AO) Method

- A simplified case of the PA method
 - Only perform offset adjustment
 - Simple pixel classification
- Classify pixels into different groups
- One offset for each group
- Details in JCTVC-D122

Fast Distortion Estimation

- Key to reduce the number of encoding passes
- Applied during rate-distortion optimization (RDO) for decision of filter parameters
- Estimate distortion without any filtering operation
- Reduce number of encoding passes from 16 to 1

Simulation Results

- JCTVC-C500 anchor
- Not only better coding efficiency but also less decoding time

	HE-AI	HE-RA	HE-LD
BD-Rate (%)	0.1	-0.4	-1.9
Encoding Time (%)	100	98	98
Decoding Time (%)	85	85	83

		HE-AI	HE-RA	HE-LD
Class A	Traffic	0.1	-0.1	
	PeopleOnStreet	0.0	0.0	
Class B	Kimono	0.3	0.2	-0.4
	ParkScene	-0.2	-0.3	-2.1
	Cactus	0.2	-0.3	-2.1
	BasketballDrive	0.1	-0.1	-0.6
Class C	BQTerrace	0.1	-0.8	-1.3
	BasketballDrill	0.2	-1.5	-3.0
	BQMall	0.2	-0.6	-2.3
	PartyScene	0.0	-0.4	-1.6
	RaceHorses	0.2	0.0	-0.5
	BasketballPass	0.0	-0.2	-0.5
Class D	BQSquare	-0.5	-1.6	-2.6
	BlowingBubble	0.0	-0.4	-1.1
	RaceHorses	0.3	-0.3	-1.1
	Vidyo1	0.4		-2.7
Class E	Vidyo3	0.3		-5.6
	Vidyo4	0.3		-2.1

Cross Verification

- We thank Intel, Panasonic, and Sharp for crosschecking our proposal
 - Intel: JCTVC-D170
 - Panasonic: JCTVC-D216
 - Sharp: JCTVC-D114
- BD-rates and run times are confirmed

Conclusions

- MQT_ALF
 - Filter Unit (FU) based local adaptation
 - Three enhancement methods – Pixel Adaptation (PA), Single Filter (SF), and Adaptive Offset (AO)
- When compared with the JCTVC-C500 anchor
 - 1-pass encoding for MQT_ALF vs. 16-pass encoding for anchor
 - Better coding efficiency and less decoding time

	HE-AI	HE-RA	HE-LD
BD-Rate	0.1%	-0.4%	-1.9%
Encoding Time Reduction	0%	2%	2%
Decoding Time Reduction	15%	15%	17%

- Propose to adopt MQT_ALF into HM for further study and improvement by all experts

Thank You.

