

JVET-P0192

CE5-related: Reducing the number of luma filters in ALF

Authors:

Olena Chubach, Ching-Yeh Chen, Tzu-Der Chuang, Yu-Wen Huang, Shaw-Min Lei

Presenter: Olena Chubach

Overall Summary

- The memory required for all ALF coefficients is significant to be stored on chip
- Propose to constrain the number of luma ALF filters in one filter set
 - Keeping flexibility of up to 25 initial luma filter classes

Over VTM-6.0 (%)		Y	U	V	EncT*	DecT*	
1	Group into 16 luma filters	AI	0.00	0.00	0.00	100%	108%
		RA	0.00	0.01	-0.01	100%	109%
		LB	0.00	0.03	0.04	100%	110%

*Runtime is inaccurate.

Introduction

- One APS includes one luma filter set (up to 25 luma filters) and up to 8 chroma filters.
- One slice can refer up to 6 APSs, and up to 8 APSs can be activated in one picture at the same time.
- The required memory storage of one APS is 435 bytes
 - $(25 \text{ luma filt.} * 12 \text{ coeff. per luma filt.} * (1 \text{ byte per coeff.} + 2 \text{ bits per clip. idx}) + 8 \text{ chroma filt.} * 6 \text{ coeff. per chroma filt.} * (1 \text{ byte per coeff.} + 2 \text{ bits per clip. idx}))$
- If ALF is applied at the decoder to each slice sequentially, the required on-chip memory size is 2610 bytes
 - $(435 \text{ bytes per APS} * 6 \text{ APSs})$
- If ALF is applied at the decoder to slices in raster scan of one picture, the required on-chip memory size becomes 3480 bytes
 - $(435 \text{ bytes per APS} * 8 \text{ APSs})$

Proposed Method

- Constrain the number of filters in one filter set by forcing luma filters to merge into up to MaxNumLumaFilt classes
- The required on-chip memory per APS can be reduced to
 - $(\text{MaxNumLumaFilt} * 12 \text{ coeff. per luma filt.} * (1 \text{ byte per coeff.} + 2 \text{ bits per clip. idx}) + 8 \text{ chroma filt.} * 6 \text{ coeff. per chroma filt.} * (1 \text{ byte per coeff.} + 2 \text{ bits per clip. idx}))$
- If MaxNumLumaFilters=16, then the required on-chip memory size can be reduced by 31%:
 - from 2610 bytes to 1800 bytes in the slice-level pipeline
 - from 3480 to 2400 bytes in the picture-level pipeline

Simulation Results

- Remove
MaxNumLumaFilters=16
- Runtime is not accurate

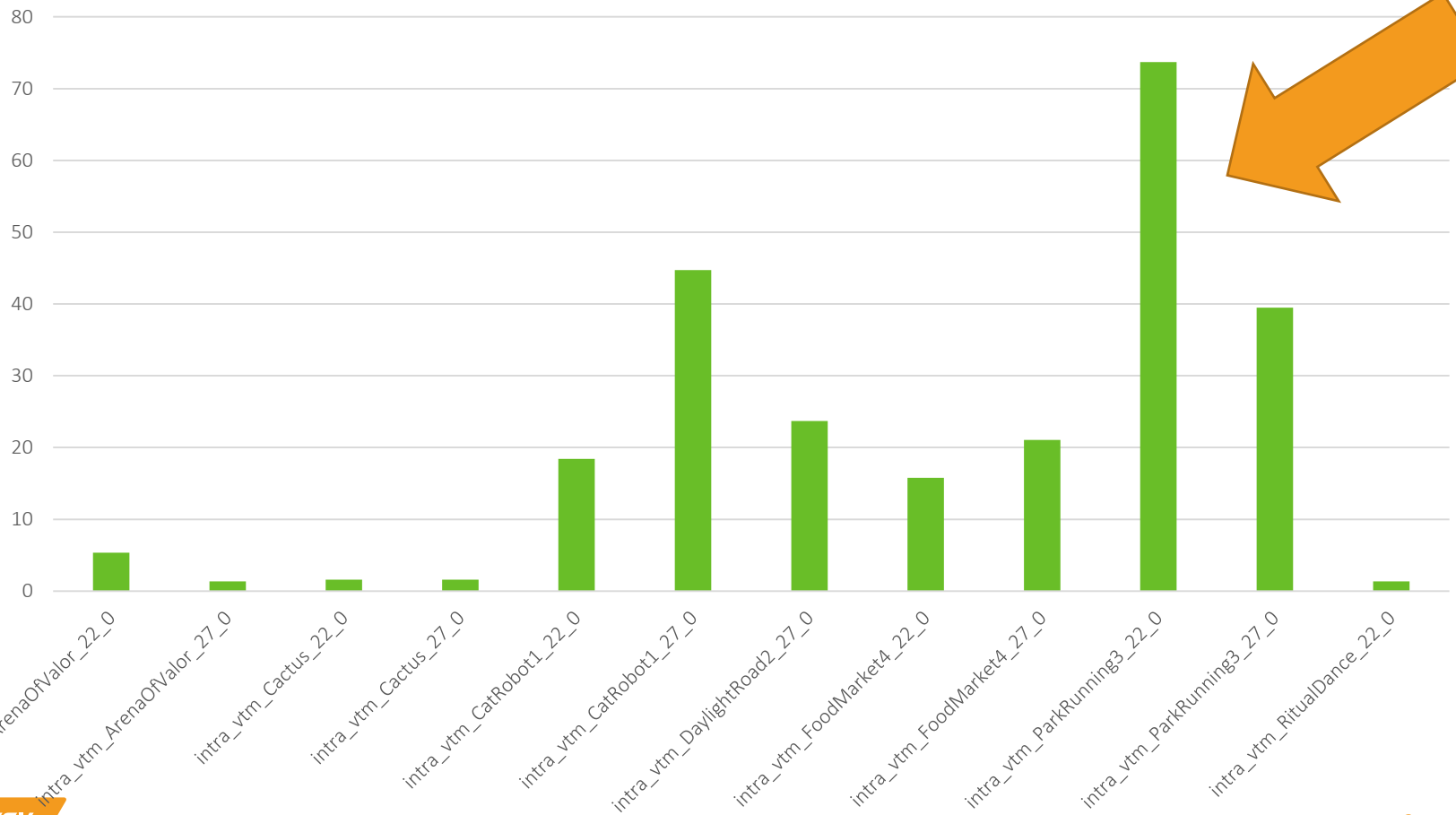
	All Intra Main10				
	Over VTM-6.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	0.00%	0.00%	100%	107%
Class A2	0.00%	0.00%	0.00%	101%	101%
Class B	0.00%	0.00%	0.00%	100%	106%
Class C	0.00%	0.00%	0.00%	100%	113%
Class E	0.00%	0.00%	0.00%	99%	113%
Overall	0.00%	0.00%	0.00%	100%	108%
Class D	0.00%	0.00%	0.00%	98%	120%
Class F	0.00%	0.00%	0.00%	100%	107%

	Random access Main10				
	Over VTM-6.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	0.01%	0.01%	100%	107%
Class A2	0.00%	0.03%	-0.03%	100%	106%
Class B	0.00%	0.00%	0.00%	100%	108%
Class C	0.00%	0.00%	0.00%	100%	113%
Class E					
Overall	0.00%	0.01%	-0.01%	100%	109%
Class D	0.00%	0.00%	0.00%	100%	117%
Class F	0.00%	0.00%	0.01%	100%	107%

	Low delay B Main10				
	Over VTM-6.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	-0.01%	0.07%	0.09%	100%	110%
Class C	0.00%	0.00%	0.00%	100%	109%
Class E	0.00%	0.00%	0.00%	101%	112%
Overall	0.00%	0.03%	0.04%	100%	110%
Class D	0.00%	0.00%	0.00%	100%	114% ⁵
Class F	0.00%	0.00%	0.00%	100%	111%

Additional analysis on number of ALF filters

- Percentage of APSs with more than 16 filters @ AI
- Average Number of filters in APS: 8



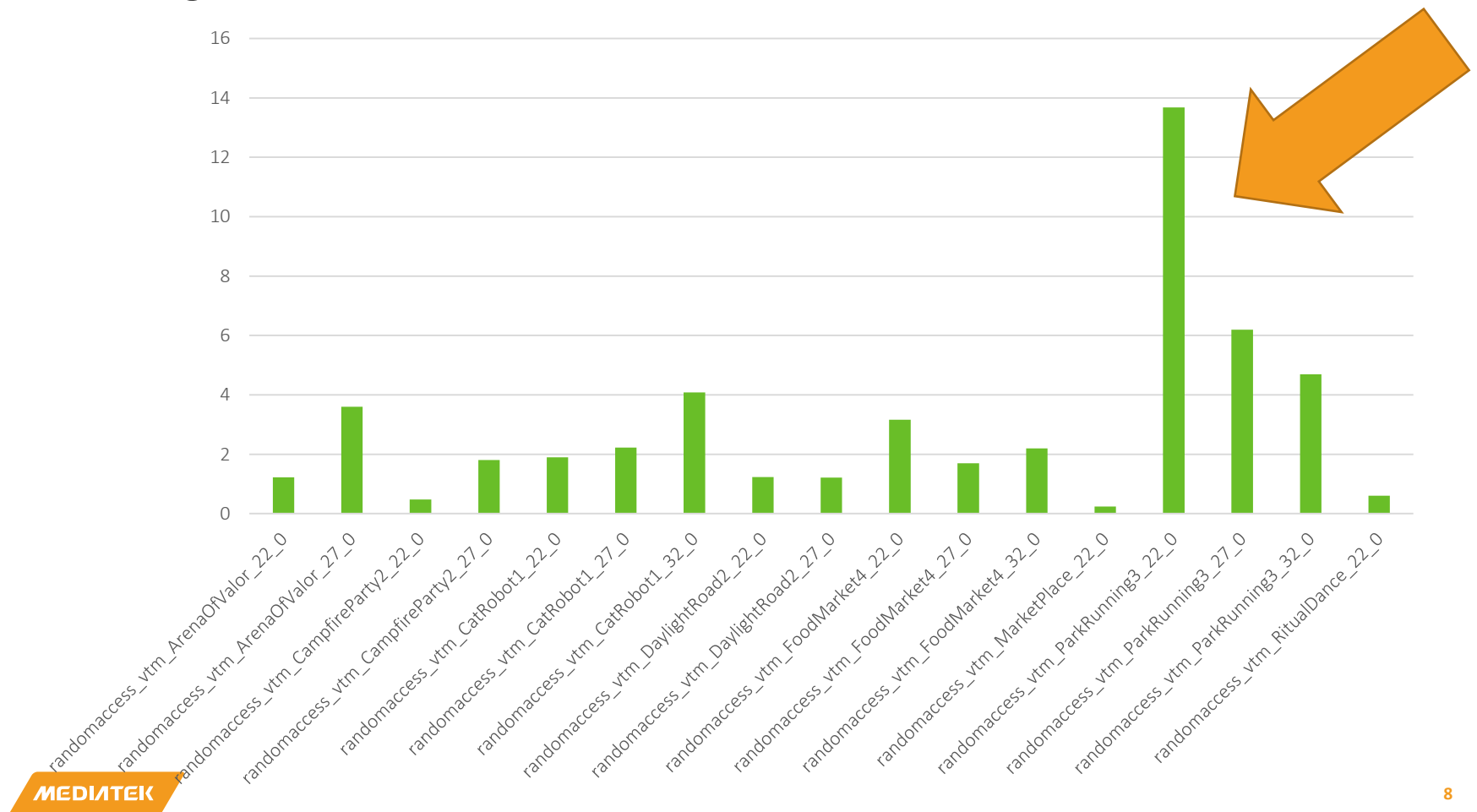
Additional analysis on number of ALF filters

- BD-rate results for ParkRunning @ AI

		Reference				Tested				BD-rate (piecewise cubic)		
	QP	kbps	Y psnr	U psnr	V psnr	kbps	Y psnr	U psnr	V psnr	Y	U	V
ParkRunning3	22	49352.40	47.08	40.27	40.79	49351.65	47.08	40.27	40.79	0.00%	0.00%	0.00%
	27	28173.03	43.23	36.69	37.65	28172.68	43.23	36.69	37.65			
	32	14887.39	39.53	33.70	34.93	14887.39	39.53	33.70	34.93			
	37	7573.12	36.06	31.61	33.11	7573.12	36.06	31.61	33.11			

Additional analysis on number of ALF filters

- Percentage of APSs with more than 16 filters @ RA
- Average Number of filters in APS: 4



Additional analysis on number of ALF filters

- BD-rate results for ParkRunning and RitualDance @ RA

		Reference				Tested				BD-rate (piecewise cubic)		
	QP	kbps	Y psnr	U psnr	V psnr	kbps	Y psnr	U psnr	V psnr	Y	U	V
ParkRunning3	22	115672.43	43.82	37.98	38.89	115656.63	43.81	37.98	38.89	0.01%	0.01%	0.00%
	27	44182.28	39.83	34.89	36.18	44185.61	39.83	34.89	36.18			
	32	19148.58	36.52	32.85	34.36	19146.74	36.52	32.85	34.36			
	37	8058.03	33.50	31.16	32.87	8058.03	33.50	31.16	32.87			
RitualDance	22	6184.43	46.33	48.26	49.65	6184.40	46.33	48.26	49.65	0.00%	0.00%	0.00%
	27	3610.57	43.51	45.95	47.20	3610.57	43.51	45.95	47.20			
	32	2024.34	40.56	43.73	44.69	2024.34	40.56	43.73	44.69			
	37	1099.05	37.60	42.13	42.64	1099.05	37.60	42.13	42.64			

Additional analysis on number of ALF filters

- Non-CTC results @ AI:
 - Class A – 225 4K images from OpenImage dataset
 - Class B – 275 FullHD images from OpenImage dataset
- Average number of filters in APS: 7
- Amount of images with more than 16 filters in APS: 2%

	All Intra Main10 Over VTM-6.0		
	Y	U	V
Class A	0.00%	0.00%	0.00%
Class B	0.00%	0.00%	0.00%
Overall	0.00%	0.00%	0.00%

Additional analysis on number of ALF filters

- Example of APSs with more than 16 filters @ AI - nonCTC



Additional analysis on number of ALF filters

- BD-rate results for ParkRunning @ AI

		Reference				Tested				BD-rate (piecewise cubic)		
QP		kbps	Y psnr	U psnr	V psnr	kbps	Y psnr	U psnr	V psnr	Y	U	V
Image_0000672	22	3033.66	47.95	52.48	51.28	3033.66	47.95	52.48	51.28	0.00%	0.00%	0.00%
	27	527.23	43.00	49.79	48.34	527.23	43.00	49.79	48.34			
	32	177.59	42.09	48.13	46.84	177.59	42.09	48.13	46.84			
	37	106.63	41.41	46.57	45.65	106.63	41.41	46.57	45.65			
Image_0000678	22	2392.76	47.97	52.49	52.87	2392.76	47.97	52.49	52.87	0.00%	0.00%	0.00%
	27	1041.07	43.79	49.14	50.51	1041.07	43.79	49.14	50.51			
	32	539.36	41.06	47.20	49.09	539.36	41.06	47.20	49.09			
	37	279.47	38.47	45.68	47.59	279.47	38.47	45.68	47.59			

Conclusions

- Sending more luma ALF filters does not guarantee improvement of coding performance, even for 4K sequences
- Proposed to reduce number of luma ALF filters by forcing merging into up to 16 classes
- Coding efficiency impact is minor
- Thanks to Ericsson for cross-checking