



Non-CE8.c.7: Single-source SAO and ALF virtual boundary processing with cross9x9

C.-Y. Chen, C.-M. Fu, C.-Y. Tsai, Y.-W. Huang, S. Lei (MediaTek)
S. Esenlik, M. Narroschke, T. Wedi (Panasonic)
I. S. Chong, M. Karczewicz (Qualcomm)



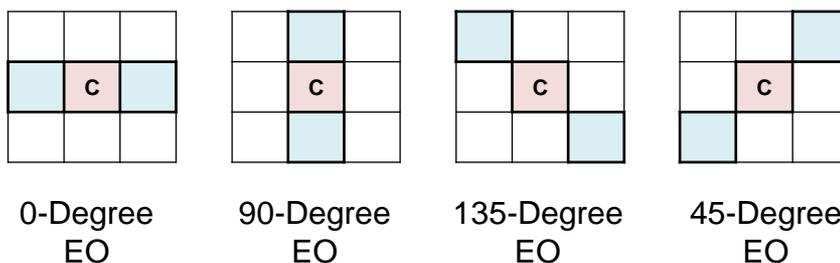
Presented by Yu-Wen Huang
7th JCT-VC Meeting in Geneva
21-30 November, 2011

Overall Summary

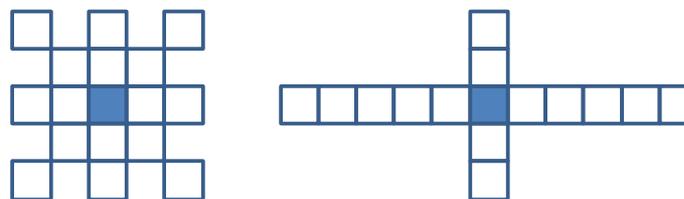
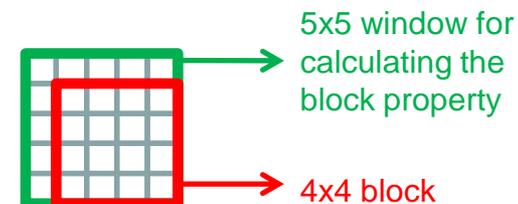
- In HM-4.0
 - SAO: 0.2 luma line, 0.2 chroma line
 - ALF: 4.1 luma lines, 4 chroma lines
- In the proposed virtual boundary (VB) processing
 - Luma VB: 4 pixels above the horizontal LCU boundary
 - Chroma VB: 2 pixels above the horizontal LCU boundary
 - Processing a pixel on one side of a VB uses reduced data from the other side of the VB
 - Does not change SAO and ALF input pixels (single-source)
 - All ALF line buffers can be saved
 - 50% SAO line buffers can be saved
 - 0-0.3% bit rate reduction
 - No noticeable run time change
 - The same visual quality
 - Suggest for adoption in HM-5.0

Background Information

- Deblocking filter (DF)
 - Luma: read 4 pixels and write 3 pixels
 - Chroma: read 2 pixels and write 1 pixel
- Sample adaptive offset (SAO)
 - Edge offset (EO): need a 3x3 window for pixel classification

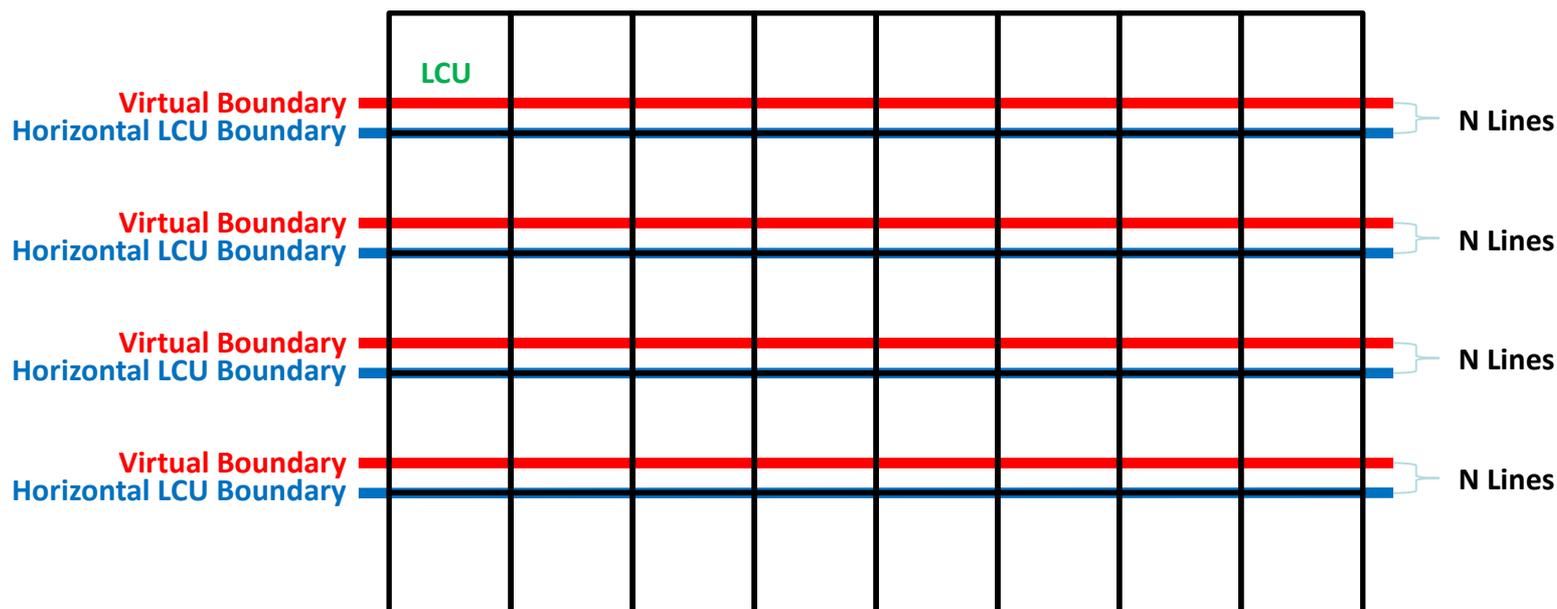


- Adaptive loop filter (ALF)
 - Calculate 4x4 block property for luma
 - Snowflake5x5
 - Cross11x5



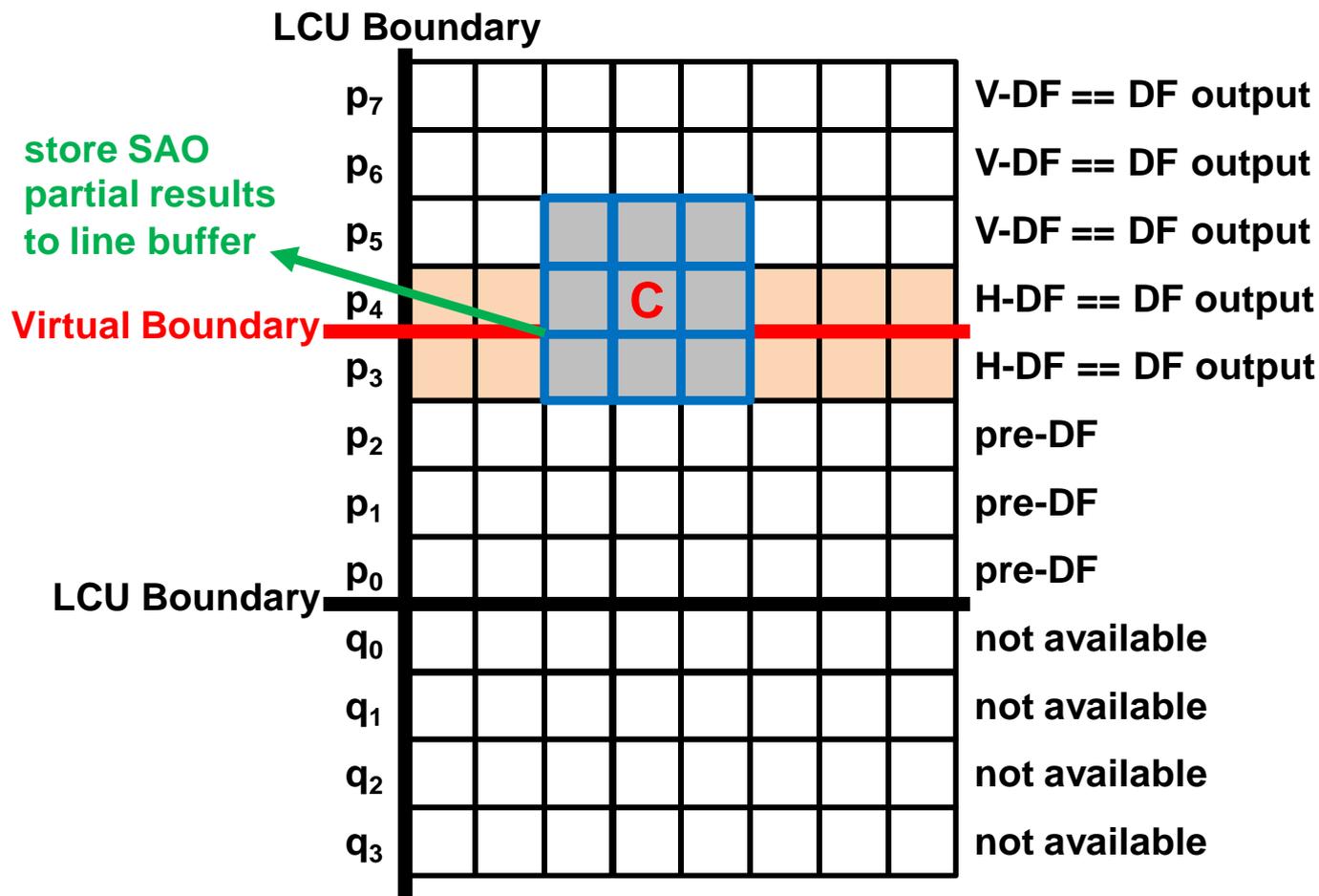
Virtual Boundary (VB)

- $N=4$ for luma, $N=2$ for chroma
- For SAO, processing a pixel on one side of a VB uses reduced data from the other side of the VB
- For ALF, processing a pixel on one side of a VB does not use any pixel on the other side of the VB



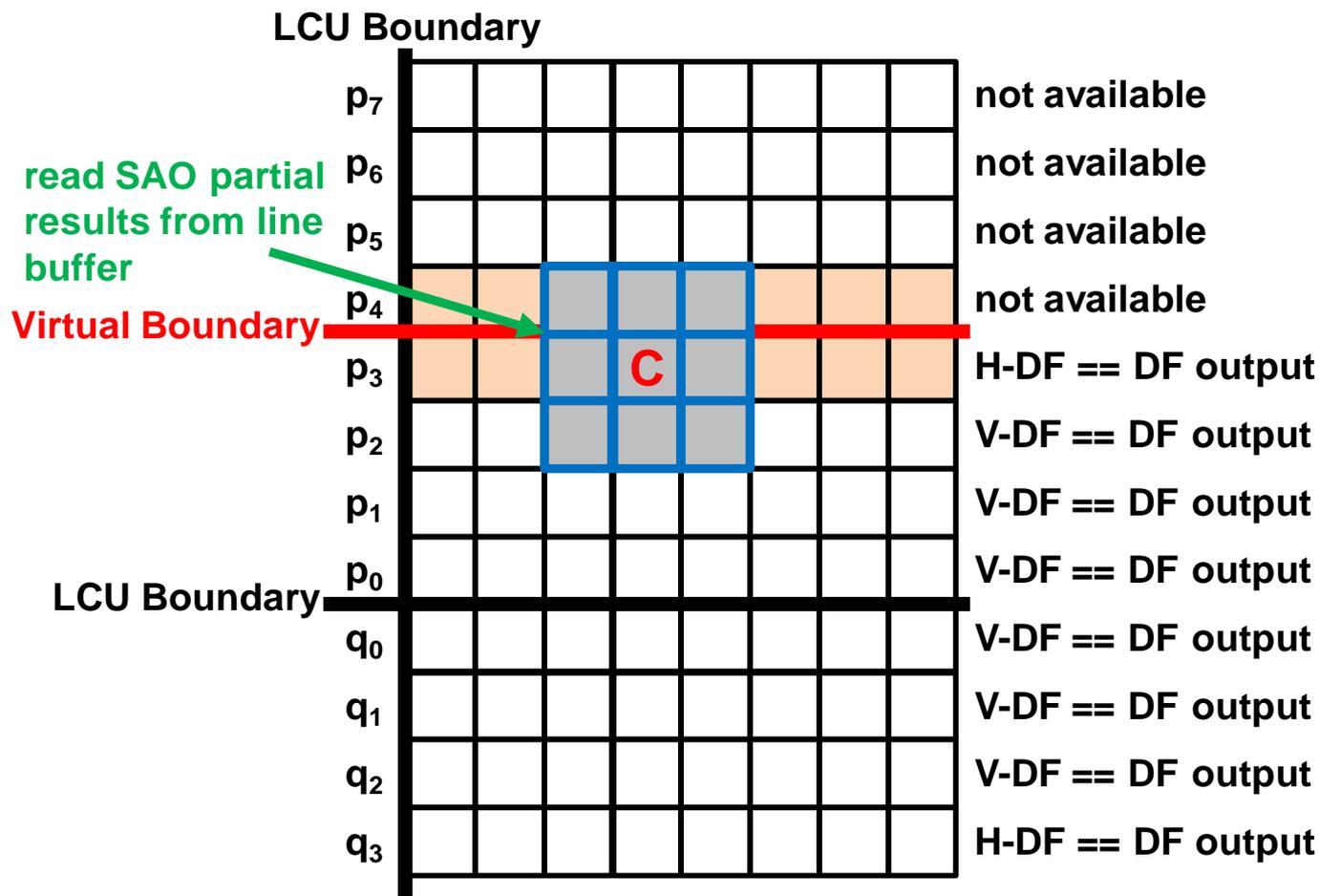
SAO VB Processing for Above-VB Pixels

- The same as HM-4.0 but compress the partial results



SAO VB Processing for Below-VB Pixels

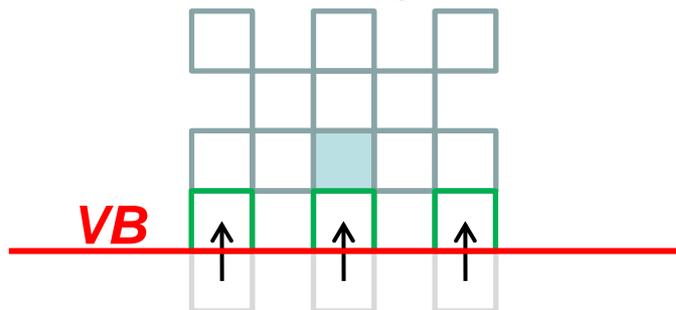
- Decompress the partial results



ALF VB Processing for Above-VB Pixels

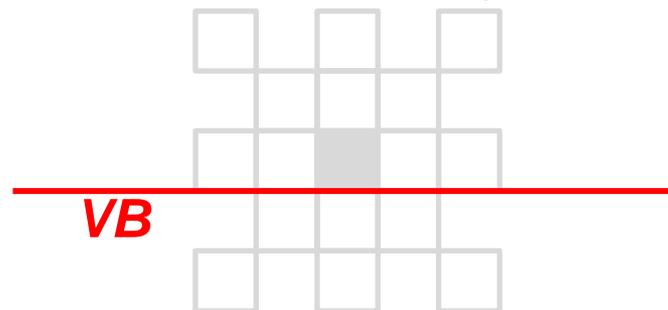
Snowflake5x5

2nd line: padding and averaging



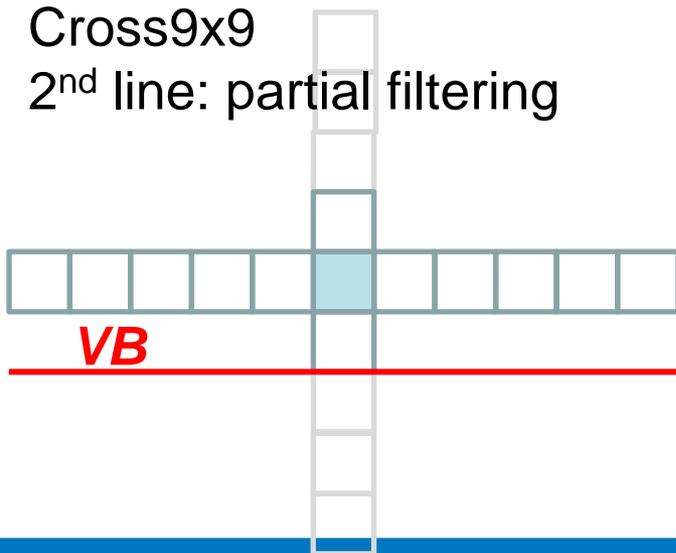
Snowflake5x5

1st line: skip filtering



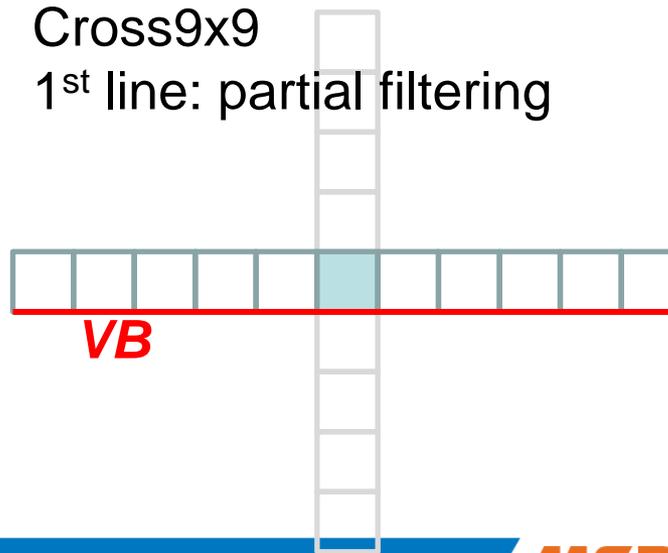
Cross9x9

2nd line: partial filtering



Cross9x9

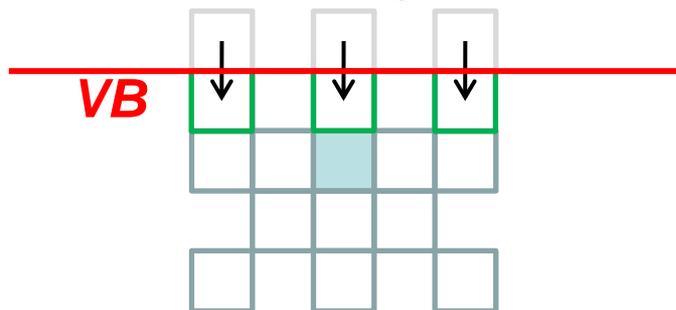
1st line: partial filtering



ALF VB Processing for Below-VB Pixels

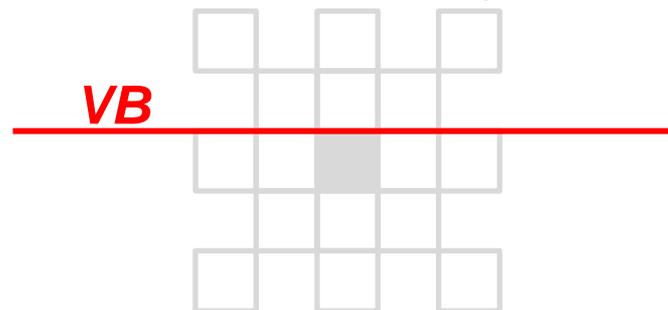
Snowflake5x5

2nd line: padding and averaging



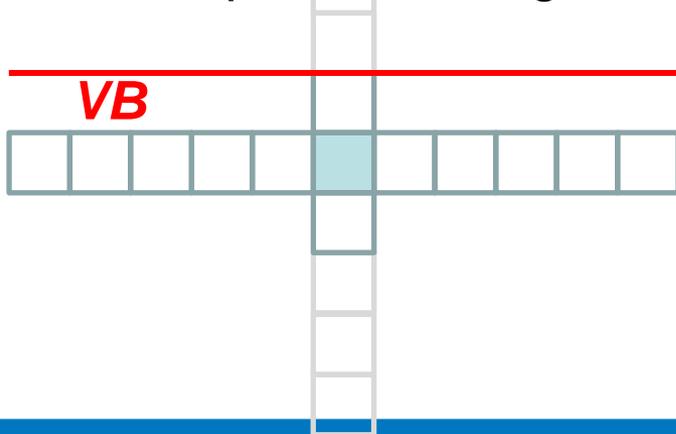
Snowflake5x5

1st line: skip filtering



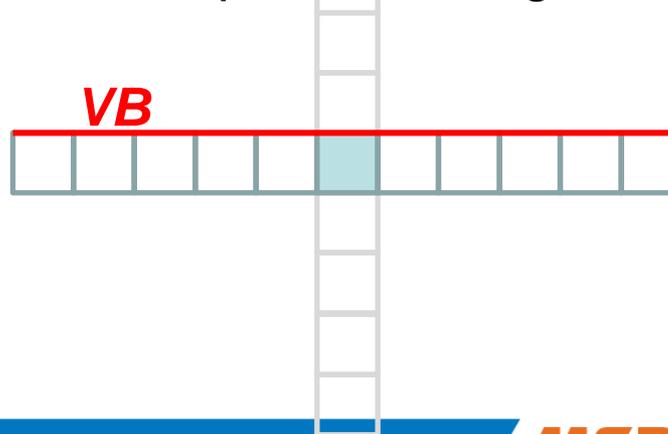
Cross9x9

2nd line: partial filtering



Cross11x5

1st line: partial filtering



Non-CE8.c.7-1 Results

- Anchor: JCTVC-F900
- Apply VB processing for ALF only
- 0.0-0.3% gain
- Similar run time
- Same visual quality

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A	0.2%	0.3%	0.2%	0.0%	0.0%	0.0%
Class B	0.0%	0.5%	0.3%	0.0%	0.0%	0.0%
Class C	0.1%	0.1%	0.3%	0.0%	0.0%	0.0%
Class D	-0.1%	0.2%	0.3%	0.0%	0.0%	0.0%
Class E	0.1%	1.0%	0.7%	0.0%	0.0%	0.0%
Overall	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%
	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%
Enc Time[%]	99%			100%		
Dec Time[%]	100%			100%		

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	0.6%	0.5%	0.0%	0.0%	0.0%
Class B	0.0%	1.0%	0.6%	0.0%	0.0%	0.0%
Class C	-0.1%	0.1%	0.1%	0.0%	0.0%	0.0%
Class D	-0.6%	0.1%	0.2%	0.0%	0.0%	0.0%
Class E						
Overall	-0.1%	0.5%	0.4%	0.0%	0.0%	0.0%
	-0.1%	0.5%	0.4%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

	Low Delay B HE			Low Delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	-0.2%	0.6%	0.5%	0.0%	0.0%	0.0%
Class C	-0.3%	0.1%	0.2%	0.0%	0.0%	0.0%
Class D	-0.8%	0.1%	-0.4%	0.0%	0.0%	0.0%
Class E	0.0%	1.2%	0.5%	0.0%	0.0%	0.0%
Overall	-0.3%	0.5%	0.2%	0.0%	0.0%	0.0%
	-0.3%	0.4%	0.1%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

	Low Delay P HE			Low Delay P LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.9%	0.8%	0.0%	0.0%	0.0%
Class C	0.0%	-0.1%	0.1%	0.0%	0.0%	0.0%
Class D	-0.1%	0.3%	0.0%	0.0%	0.0%	0.0%
Class E	0.0%	1.3%	0.3%	0.0%	0.0%	0.0%
Overall	-0.1%	0.6%	0.3%	0.0%	0.0%	0.0%
	0.0%	0.5%	0.3%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

Non-CE8.c.7-2 Results

- Anchor: JCTVC-F900
- Apply VB processing for both SAO and ALF
- 0.0-0.3% gain
- Similar run time
- Same visual quality

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A	0.2%	0.3%	0.2%	0.0%	0.0%	0.0%
Class B	0.0%	0.5%	0.3%	0.0%	0.0%	0.0%
Class C	0.1%	0.1%	0.3%	0.0%	0.0%	0.0%
Class D	-0.1%	0.2%	0.3%	0.0%	0.0%	0.0%
Class E	0.1%	1.0%	0.8%	0.0%	0.1%	0.0%
Overall	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%
	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%
Enc Time[%]	99%			99%		
Dec Time[%]	100%			100%		

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.1%	0.6%	0.6%	-0.1%	0.1%	0.0%
Class B	0.0%	0.9%	0.6%	0.0%	0.0%	0.1%
Class C	-0.1%	0.0%	0.2%	0.0%	0.0%	0.0%
Class D	-0.6%	0.1%	0.2%	0.0%	0.0%	0.0%
Class E						
Overall	-0.1%	0.4%	0.4%	0.0%	0.0%	0.0%
	-0.1%	0.4%	0.4%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	99%			101%		

	Low Delay B HE			Low Delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	-0.2%	0.6%	0.3%	0.0%	0.1%	0.2%
Class C	-0.3%	-0.1%	-0.1%	0.0%	0.1%	-0.1%
Class D	-0.8%	-0.1%	0.1%	0.0%	0.3%	-0.1%
Class E	0.0%	0.6%	0.7%	0.0%	0.4%	-0.5%
Overall	-0.3%	0.2%	0.2%	0.0%	0.2%	-0.1%
	-0.3%	0.2%	0.1%	0.0%	0.2%	-0.1%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

	Low Delay P HE			Low Delay P LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.9%	1.0%	0.0%	0.1%	0.1%
Class C	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%
Class D	-0.1%	-0.1%	-0.1%	0.0%	-0.2%	-0.2%
Class E	0.1%	0.9%	-0.1%	0.0%	0.4%	0.6%
Overall	0.0%	0.4%	0.3%	0.0%	0.1%	0.1%
	0.0%	0.4%	0.2%	0.0%	0.0%	0.1%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			100%		

Conclusion

- Combine single-source SAO and ALF VB processing and snowflake5x5+cross9x9
- Non-CE8.c.7-1
 - VB processing for ALF only
 - Remove all ALF line buffers
 - 0-0.3% coding efficiency gain
 - Same visual quality
- Non-CE8.c.7-2
 - VB processing for both SAO and ALF
 - Remove all ALF line buffers and 50% SAO line buffers
 - 0-0.3% coding efficiency gain
 - Same visual quality