



CE8.c.3: Multi-source SAO and ALF virtual boundary processing

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



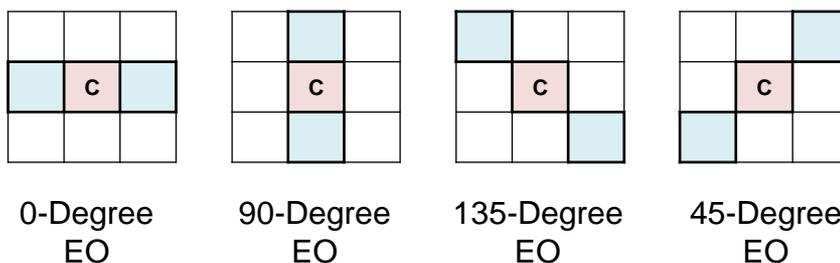
Presented by Yu-Wen Huang
7th JCT-VC Meeting in Geneva
21-30 November, 2011

Overall Summary

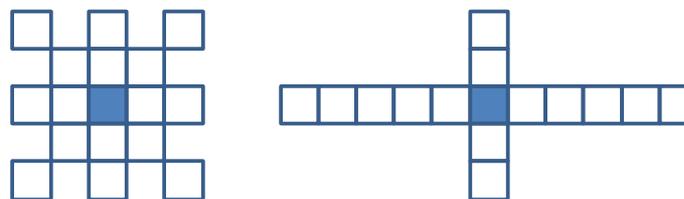
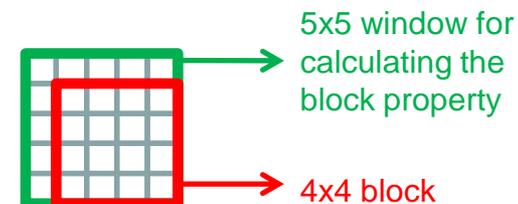
- In HM-4.0
 - SAO: 0.2 luma line, 0.2 chroma line
 - ALF: 4.1 luma lines, 4 chroma lines
- In the proposed virtual boundary (VB) processing
 - Luma VB: 4 pixels above the horizontal LCU boundary
 - Chroma VB: 2 pixels above the horizontal LCU boundary
 - Processing a pixel on one side of a VB does not use any pixel on the other side of the VB unless it is already in the buffer
 - Use pre-DF pixels below the VB when processing pixels above the VB
 - Change SAO and ALF input pixels (multi-source)
 - All SAO and ALF line buffers can be saved
 - 0.1-0.2% bit rate increase
 - No noticeable encoding time change
 - 2-3% decoding time increase
 - Very slightly degraded visual quality

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

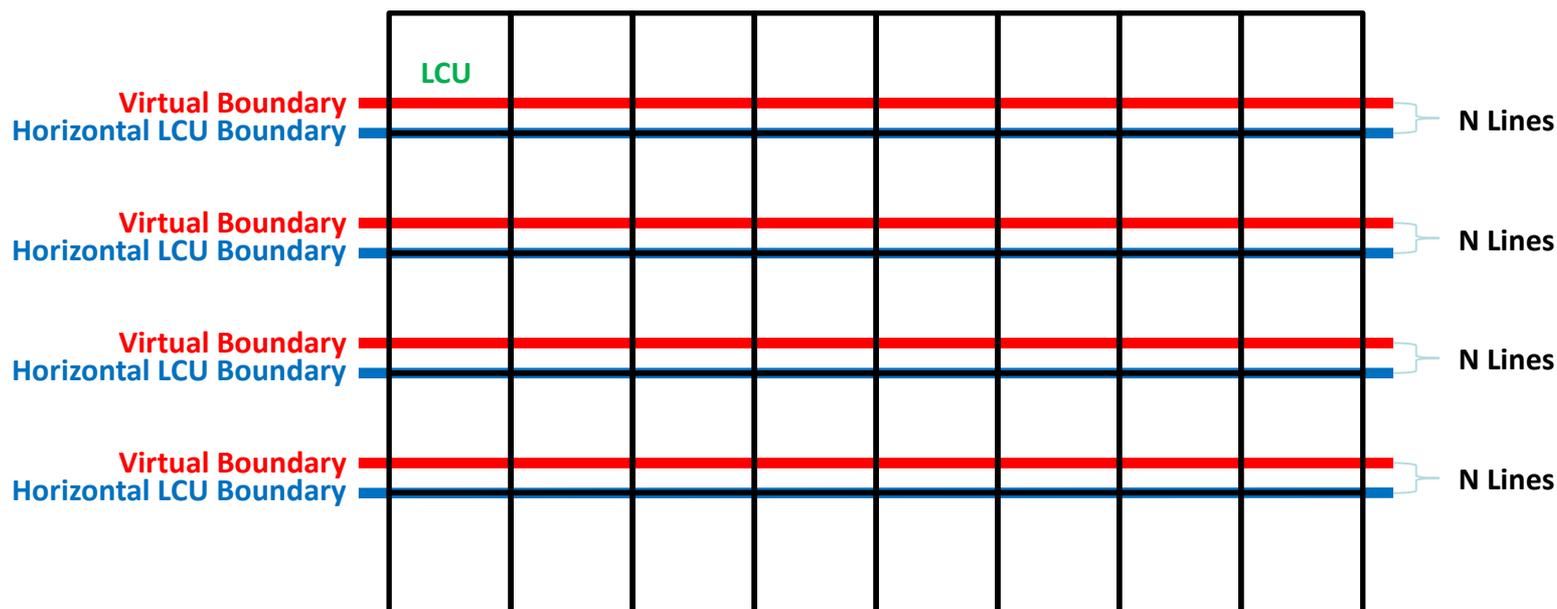


- Adaptive loop filter (ALF)
 - Calculate 4x4 block property for luma
 - Snowflake5x5
 - Cross11x5

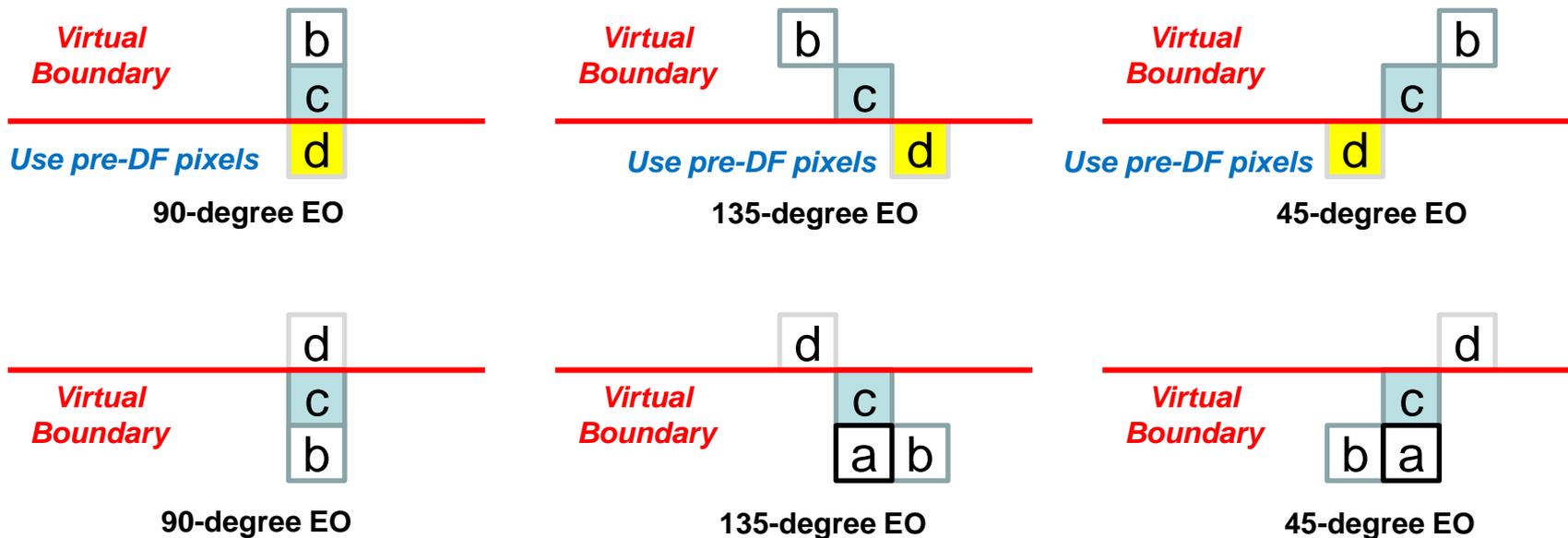


Virtual Boundary (VB)

- $N=4$ for luma, $N=2$ for chroma
- Processing a pixel on one side of a VB does not use any pixel on the other side of the VB unless it is already in the buffer
- Allow to use pre-DF pixels below the VB when pixels above the VB are processed



SAO VB Processing



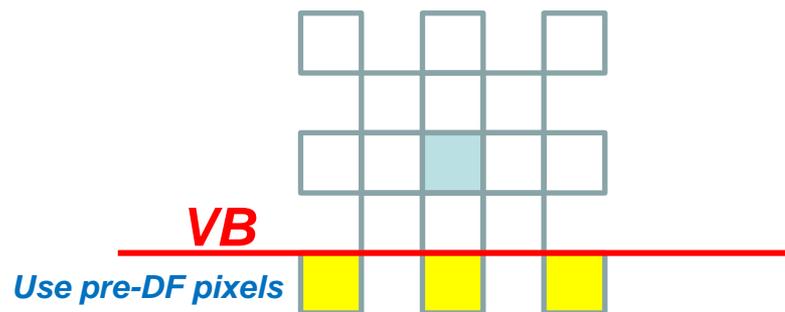
```

if ( ( |c-a| < TH ) && (sanity check is passed) )
    c uses b's offset
else
    c uses zero offset
  
```

ALF VB Processing for Above-VB Pixels

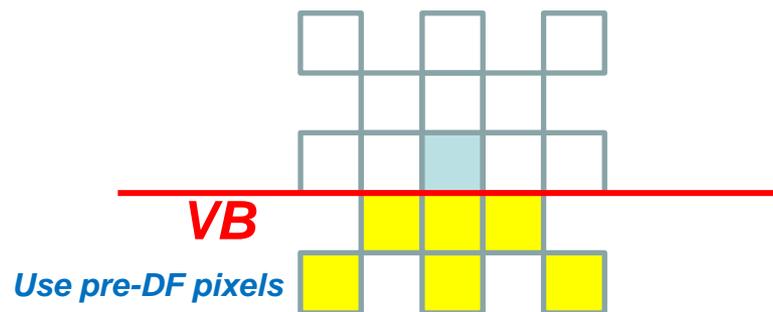
Snowflake5x5

2nd line: multi-source



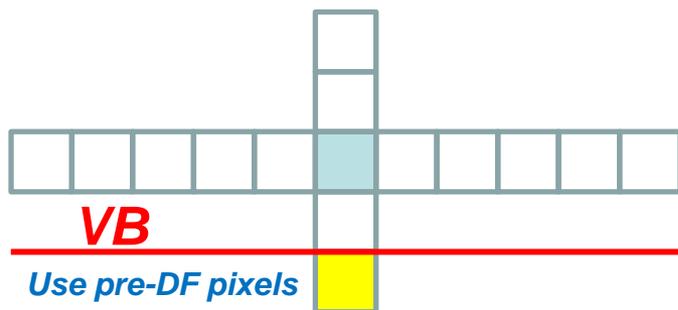
Snowflake5x5

1st line: multi-source



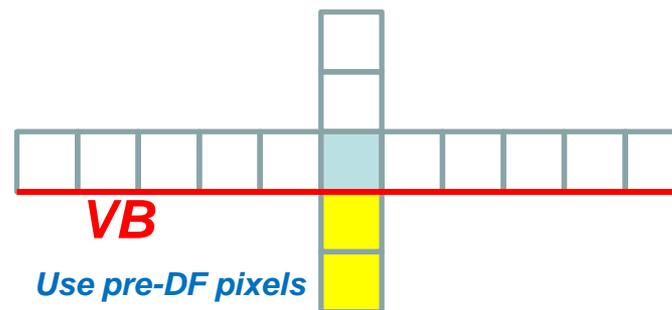
Cross11x5

2nd line: multi-source



Cross11x5

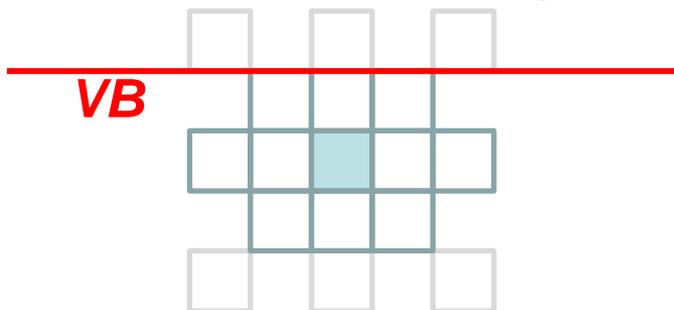
1st line: multi-source



ALF VB Processing for Below-VB Pixels

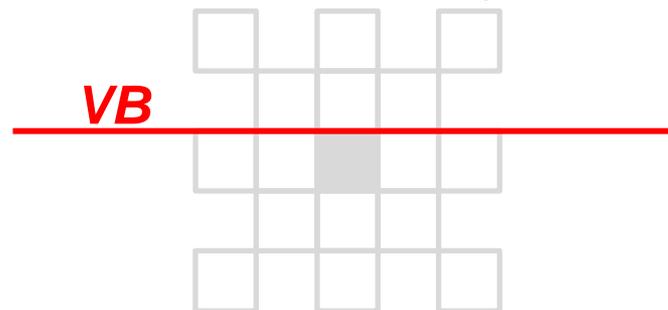
Snowflake5x5

2nd line: partial filtering



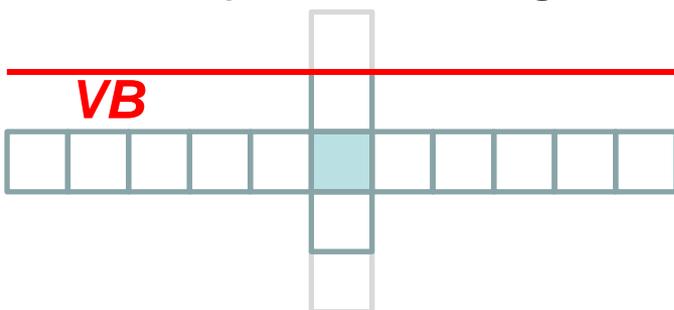
Snowflake5x5

1st line: skip filtering



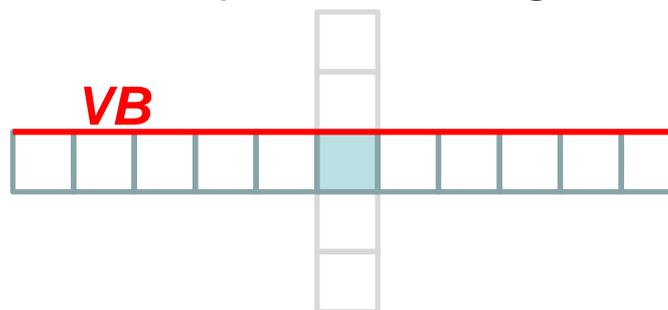
Cross11x5

2nd line: partial filtering



Cross11x5

1st line: partial filtering



Simulation Results

- Anchor: JCTVC-F900
- 0.1-0.2% loss in HE
- 0.0-0.1% loss in LC
- Similar encoding time
- 2-3% decoding time increase
- Very slightly degraded visual quality in HE
- Same visual quality in LC

	All Intra HE			All Intra LC		
	Y	U	V	Y	U	V
Class A	0.2%	0.4%	0.4%	0.0%	0.1%	0.1%
Class B	0.1%	0.5%	0.3%	0.0%	0.1%	0.1%
Class C	0.0%	0.2%	0.3%	0.0%	0.1%	0.1%
Class D	0.0%	0.2%	0.3%	0.0%	0.0%	0.1%
Class E	0.1%	1.0%	0.8%	0.0%	0.1%	0.1%
Overall	0.1%	0.4%	0.4%	0.0%	0.1%	0.1%
	0.1%	0.4%	0.4%	0.0%	0.1%	0.1%
Enc Time[%]	99%			99%		
Dec Time[%]	102%			102%		

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.3%	0.5%	0.6%	0.0%	0.1%	0.2%
Class B	0.1%	1.0%	0.7%	0.0%	0.1%	0.1%
Class C	0.0%	0.2%	0.4%	0.0%	-0.1%	0.0%
Class D	0.1%	0.4%	0.3%	0.0%	-0.1%	0.0%
Class E						
Overall	0.1%	0.6%	0.5%	0.0%	0.0%	0.1%
	0.1%	0.6%	0.5%	0.0%	0.0%	0.1%
Enc Time[%]	100%			100%		
Dec Time[%]	103%			103%		

	Low Delay B HE			Low Delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.1%	0.8%	0.6%	0.1%	0.2%	0.2%
Class C	0.0%	0.1%	0.3%	0.0%	0.3%	0.0%
Class D	0.0%	0.8%	0.6%	0.0%	0.3%	-0.1%
Class E	0.5%	0.6%	1.2%	0.0%	0.6%	0.3%
Overall	0.1%	0.6%	0.6%	0.0%	0.3%	0.1%
	0.1%	0.5%	0.6%	0.0%	0.3%	0.1%
Enc Time[%]	100%			100%		
Dec Time[%]	103%			102%		

	Low Delay P HE			Low Delay P LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.2%	1.3%	1.1%	0.1%	0.4%	0.1%
Class C	0.1%	0.3%	0.4%	0.1%	0.1%	0.2%
Class D	0.1%	0.1%	0.3%	0.0%	0.3%	-0.2%
Class E	0.5%	1.1%	0.6%	0.1%	0.6%	-0.2%
Overall	0.2%	0.7%	0.6%	0.1%	0.4%	0.0%
	0.2%	0.7%	0.6%	0.1%	0.3%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	102%			103%		

Conclusion

- Proposed virtual boundary processing for SAO and ALF
 - Remove all line buffers
 - 0.1-0.2% coding efficiency loss
 - Very slightly degraded visual quality