

# **CE8 Subtest 5: Luma ALF with reduced vertical filter size**

**(JCTVC-E060)**

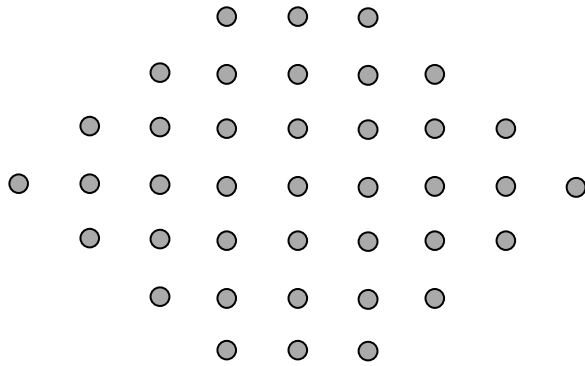
**Madhukar Budagavi, Vivienne Sze, Minhua Zhou**

**Texas Instruments**

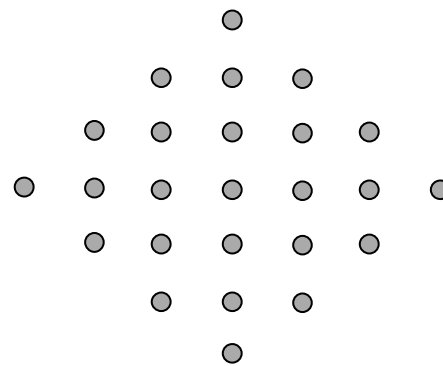
**Joint Collaborative Team on Video Coding (JCT-VC)  
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**

**5th Meeting: Geneva, CH, 16-23 March, 2011**

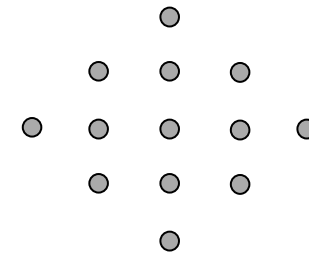
# HM 2.0 Luma ALF



9x7-Diamond



7-Diamond



5-Diamond

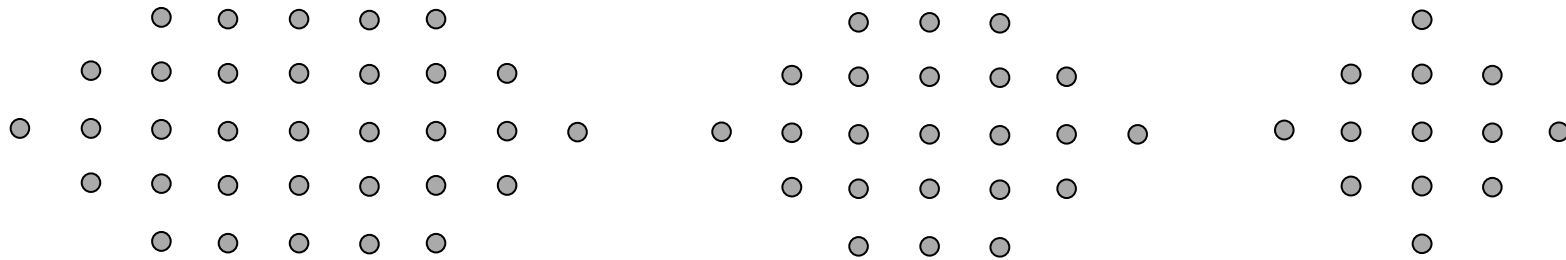
## Nx7 filter set

- HM 2.0 Luma ALF uses three diamond shaped filters with maximum vertical size of 7
- For luma, a maximum of up to 16 filters can be signaled per slice

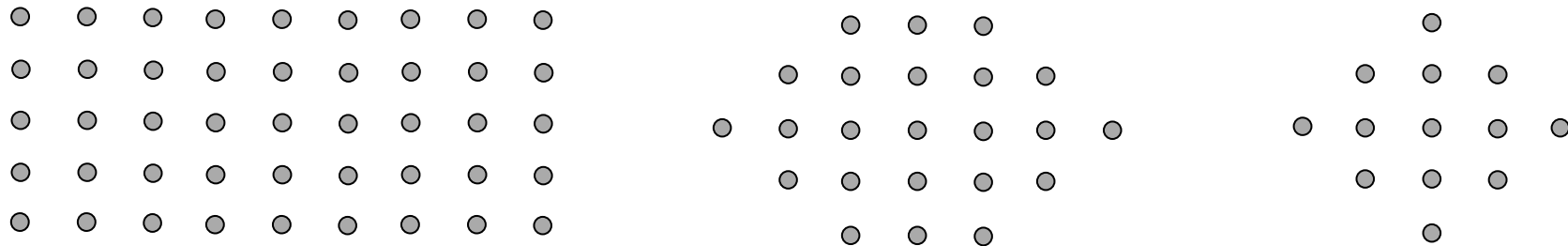
# ALF decode complexity

- Complexity considerations
  - Computational complexity
  - Memory bandwidth
  - Memory size (line buffer)
- ALF filtering can be carried out in either frame-based or LCU-based fashion
- For a given image size, the vertical size of ALF filters determines the size of line buffer/memory bandwidth requirements
  - Reducing vertical size of filter reduces line buffer/memory bandwidth requirements

# Luma ALF filters with reduced vertical size (max vertical size = 5)



Nx5-Set1



Nx5-Set3

# Results (v/s HM 2.0 anchor)

			Intra		Random access		Low delay	
	Line buffer size / Memory bandwidth	Worst case multiplies	BD-Rate	Dec time	BD-Rate	Dec time	BD-Rate	Dec time
No ALF	0	0	3.2	62%	4.9	69%	4.4	73%
HM 2.0 ALF	1X	20						
Nx5-Set 1	0.66X	17	0.1	95%	0.2	94%	0.3	96%
Nx5-Set 3	0.66X	23	0.1	99%	0.2	96%	0.3	97%

Table 1: Summary of BD-Rate, Decoder time, and complexity of ALF filters. Compared to anchor.

# Conclusions

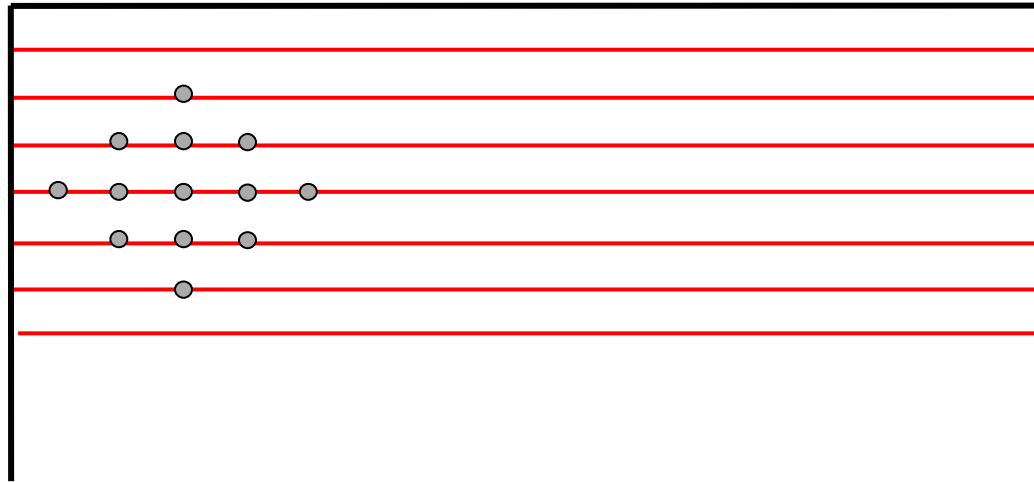
- Implementation complexity analysis involves not just analysis of computations but also analysis of memory bandwidth and memory size (area).
- For a given image size, the vertical size of ALF filters determines the size of line buffer/memory bandwidth requirements
  - Reducing vertical size of filter reduces line buffer/memory bandwidth requirements
- Contribution proposes ALF filter sets with reduced vertical size
- Proposed ALF filter sets capture most of the ALF coding gains
- Nx5-Set1 successfully cross-verified by eBrisk Video (JCTVC-Exxx). Thanks to eBrisk Video.
- Recommend that Nx5-Set1 ALF filter be adopted in HM 3.0 if there are no objections

# Results (v/s No ALF)

			Intra		Random access		Low delay	
	Line buffer size / Memory bandwidth	Worst case multiplies	BD-Rate	Dec time	BD-Rate	Dec time	BD-Rate	Dec time
No ALF	0	0		62%		69%		73%
HM 2.0 ALF	1X	20	-3.1		-4.6		-4.2	
Nx5-Set 1	0.66X	17	-2.9	95%	-4.4	94%	-3.9	96%
Nx5-Set 3	0.66X	23	-2.9	99%	-4.4	96%	-3.9	97%

Table 2: Summary of BD-Rate, Decoder time, and complexity of ALF filters. Compared to no ALF.

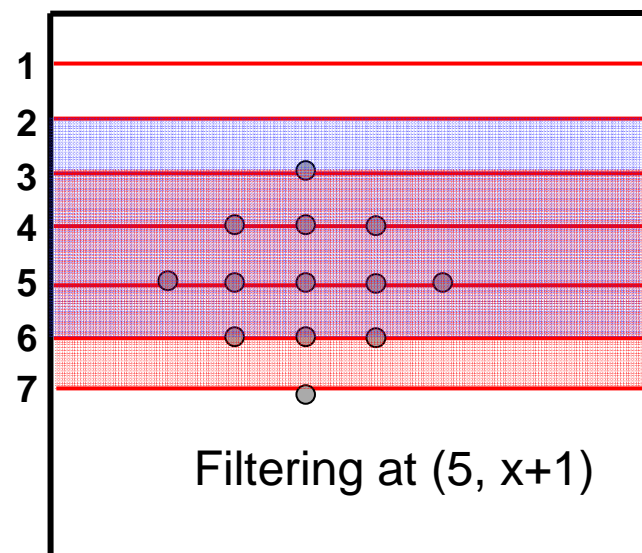
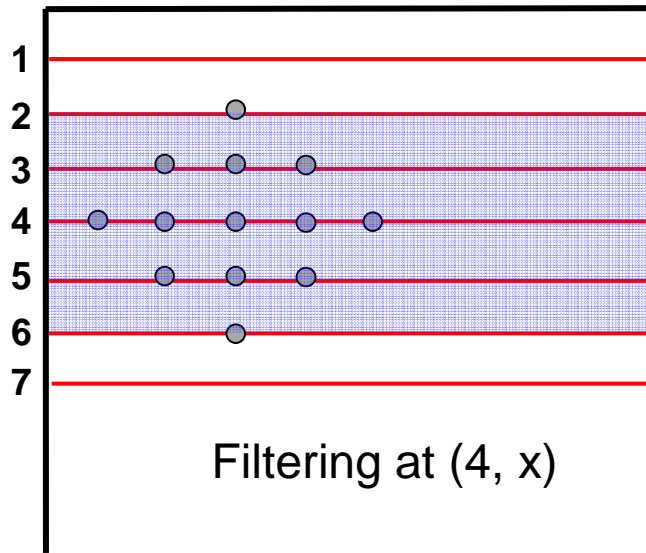
# Frame based ALF decode (No line buffers)



- For  $N = 9$ , number of input pixel reads for each filtering operation = 39
- Memory bandwidth implications
  - Deblock output frame buffer needs to be read 39 times (also written to 1 time)
  - Not practical
- Memory requirements: 1 frame buffer for storing deblock output



# Frame based ALF decode (with line buffers)

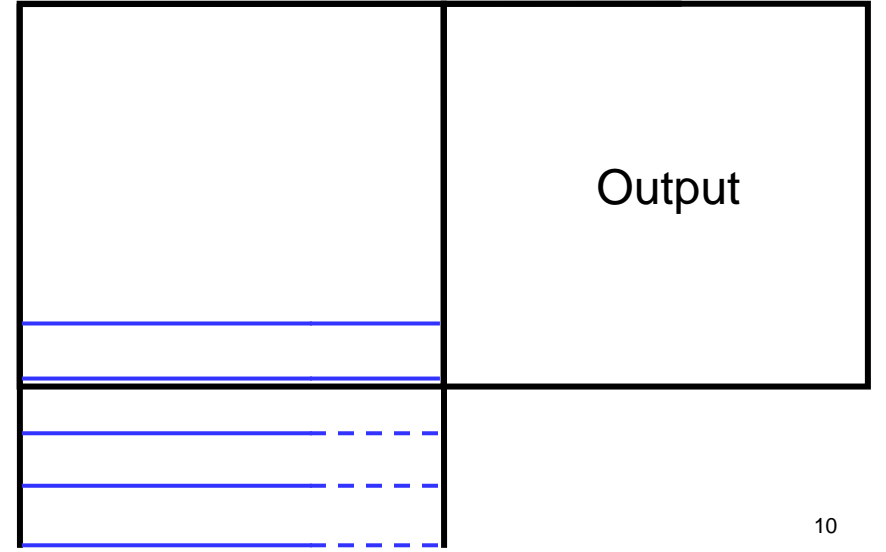
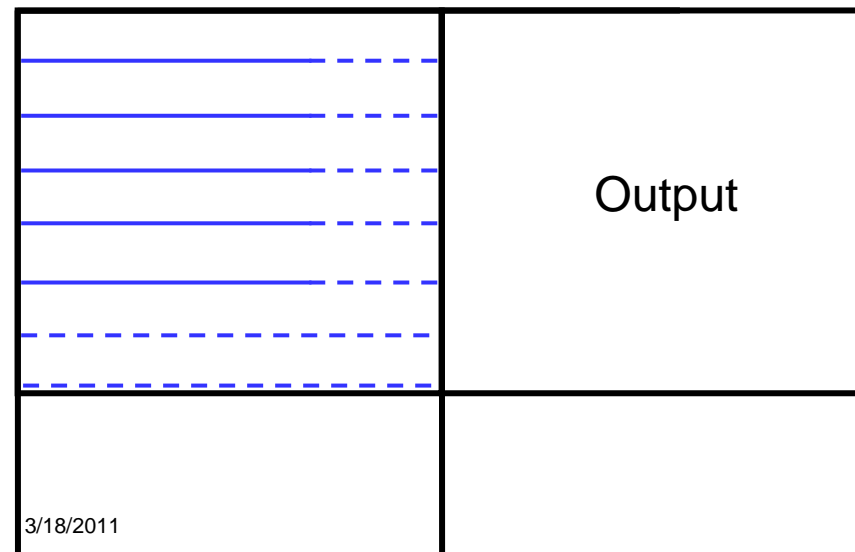
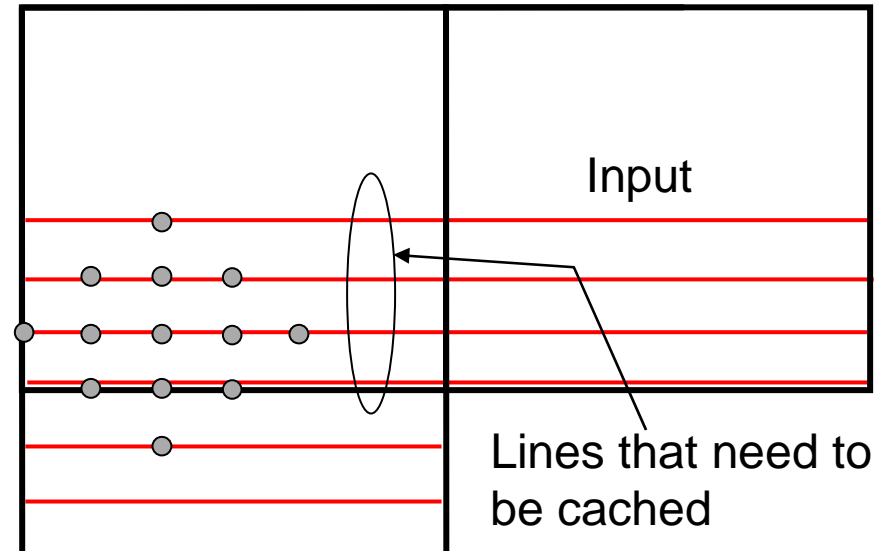
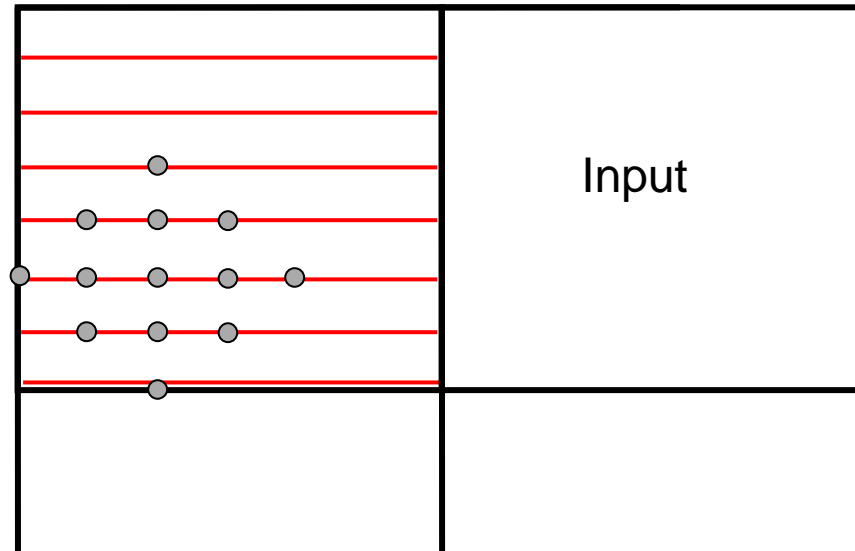


- Overlapping input data from one line to next need not be fetched when internal line buffers are used
- For  $N = 9$ , number of input pixel reads for each filtering operation = 1
- Memory bandwidth implications
  - Deblock output frame buffer needs to be read 1 time (also written to 1 time)
  - Still very expensive
- Memory requirements:  $\sim 6 \cdot \text{picWidth}$  local memory, + 1 frame buffer for storing deblock output

3/18/2011

9

# LCU-based ALF decode with line buffers



3/18/2011

10

# Summary of ALF implementation architecture

