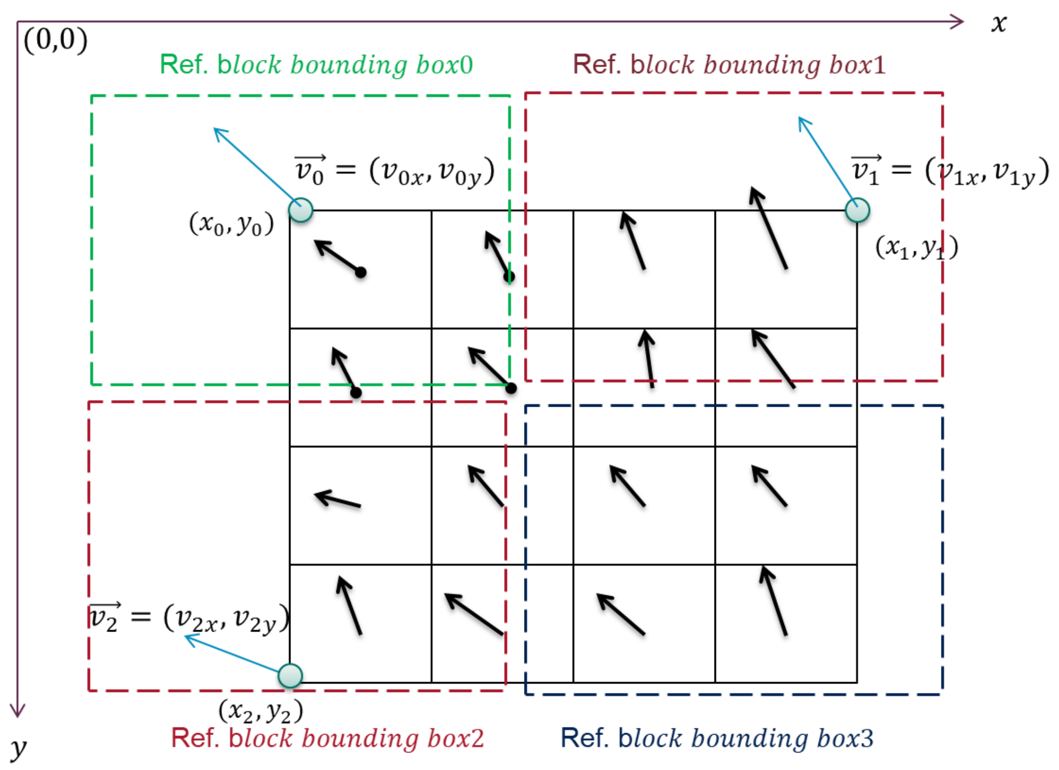
A RESTRICTION ON MEMORY BANDWIDTH CONSUMPTION OF AFFINE MODE (JVET-M0049)

Minhua Zhou

Broadcom Inc.

Background

- In the current design, 4x4 sub-block size is used for the generation of sub-block vectors of affine mode
- The sub-block motion vector spread (i.e. how far the sub-block motion vectors can go apart from each other) needs to be constrained to avoid non-affordable peak memory bandwidth consumption of the affine mode.
- Most of methods being studied just constrain the sub-block motion vector spread for each 8x8 region of a PU
  - Within the PU the sub-block motion vectors can still go widely apart from one 8x8 region to the next 8x8 region
- Propose to impose the constraint at the PU level, not just at 8x8 level.



Proposed Method

- Directly constrain the affine motion model parameters
- Have the encoder not select the model parameters *(a,b,c,d)* that lead to the reference block bounding box exceeding a pre-defined threshold
- *Method*
  - Affine motion model $\begin{cases} v_x = ax + cy + e \\ v_y = bx + dy + f \end{cases}$
  - The reference block bounding box size $(\mathrm{bxW4}, \mathrm{bxH4})$ of the bi-directional affine mode

$$\begin{cases} bxW4 = \max(0, 4(1+a), 4c, 4(1+a)+4c) - \min(0, 4(1+a), 4c, 4(1+a)+4c\ ) + 11 \\ bxH4 = \max\big(0, 4b, 4(1+d), 4b+4(1+d)\big) - \min\big(0, 4b, 4(1+d), 4b+4(1+d)\big) + 11 \end{cases}$$

  - The reference block bounding box size (bxW_h, bxH_h) of the unidirectional affine mode

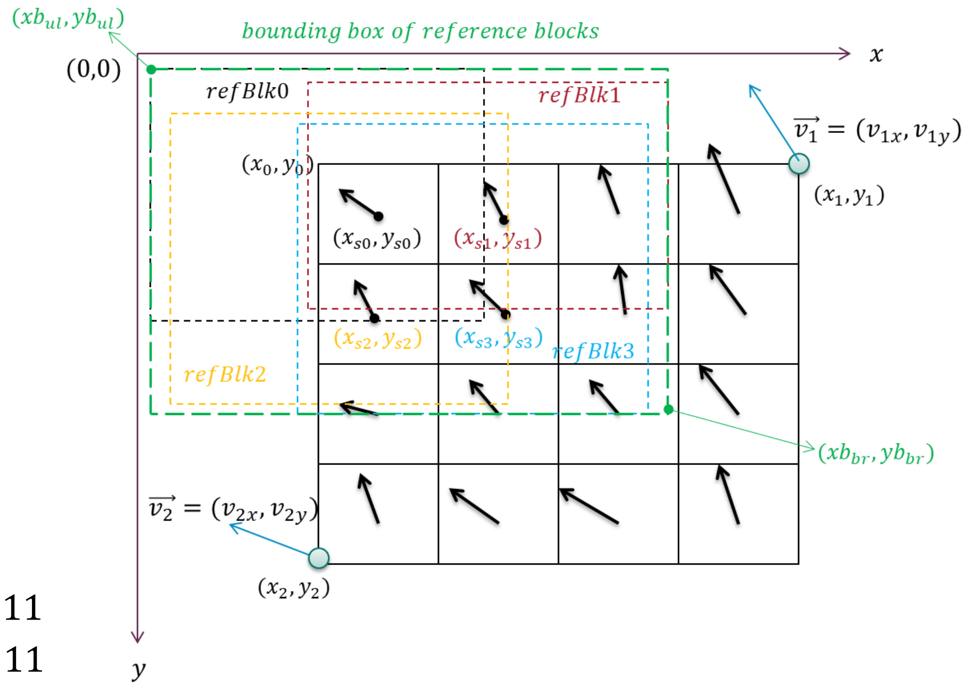
$$\begin{cases} bxW_h = \max\big(0, 4(1+a)\big) - \min\big(0, 4(1+a)\big) + 11 \\ bxH_h = \max(0, 4b\ ) - \min(0, 4b\ ) + 11 \end{cases}$$

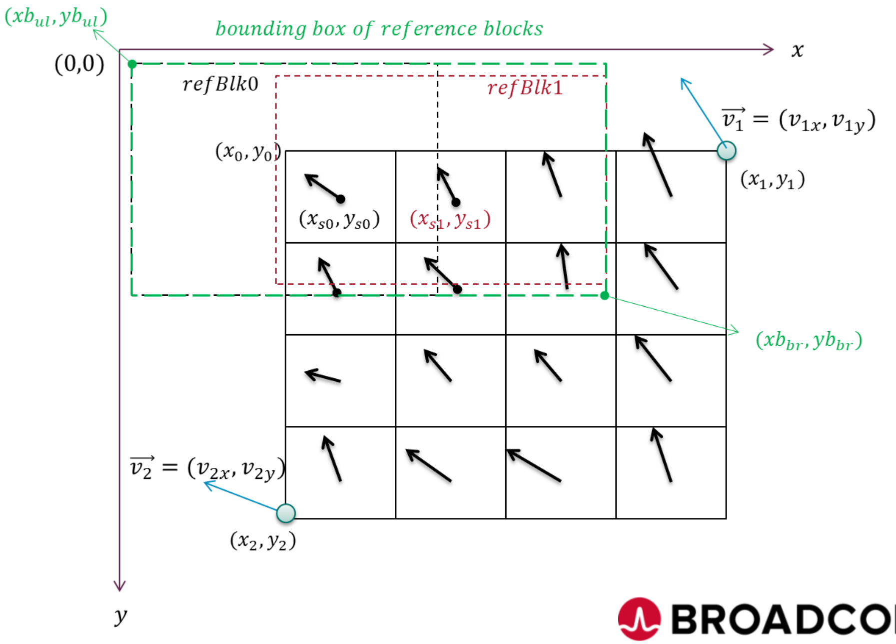
  - Pre-defined thresholds

$$\begin{cases} Thred4 = (15+\delta_x) * (15+\delta_y) \\ Thred_h = (15+\delta_x) * (11+\delta_y) \end{cases}$$

  - Constraint: a conforming stream shall satisfy

$$\begin{cases} bxW_4 * bxH_4 \le Thred_4 & for\ bi-directional\ affine\ mode \\ bxW_h * bxH_h \le Thred_h & for\ unidirectional\ affine\ mode \end{cases}$$





Experimental Results

- Tests carried out:
  1) Test 1: set threshold of luma reference block bounding box to 16*16 for 2x2 sub-block vectors of bi-directional affine mode, and to 16*12 for 2x1 sub-block vectors of unidirectional affine mode (i.e. $setting\ \delta_x = \delta_y = 1$).
  2) Test 2: set threshold of luma reference block bounding box to 17*17 for 2x2 sub-block vectors of bi-directional affine mode, and to 17*13 for 2x1 sub-block vectors of unidirectional affine mode (i.e. $setting\ \delta_x = \delta_y = 2$).
- Summary results:

Test 1

	Random Access Main 10				
	Over VTM3.0				
	Y	U	V	EncT	DecT
Class A1	0.06%	0.07%	0.08%	109%	105%
Class A2	0.10%	0.16%	0.18%	95%	100%
Class B	0.03%	0.06%	0.06%	98%	96%
Class C	0.04%	0.02%	0.09%	101%	101%
Class E					
Overall	0.05%	0.07%	0.09%	100%	100%
Class D	0.00%	-0.04%	-0.03%	99%	100%
Class F (optional)	0.05%	0.04%	0.12%	92%	92%

	Low delay B Main10				
	Over VTM3.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.04%	-0.17%	0.03%	98%	102%
Class C	0.05%	0.13%	0.02%	96%	93%
Class E	0.02%	0.51%	0.18%	105%	104%
Overall	0.04%	0.10%	0.06%	99%	99%
Class D	-0.01%	0.11%	-0.17%	98%	97%
Class F (optional)	0.02%	-0.22%	-0.17%	102%	99%

Test 2

	Random Access Main 10				
	Over VTM-3.0				
	Y	U	V	EncT	DecT
Class A1	0.00%	-0.09%	0.03%	99%	100%
Class A2	0.01%	0.03%	0.07%	97%	109%
Class B	0.01%	0.06%	-0.06%	101%	101%
Class C	0.00%	0.02%	0.07%	100%	99%
Class E					
Overall	0.01%	0.01%	0.02%	100%	102%
Class D	0.01%	-0.10%	0.03%	91%	87%
Class F (optional)	0.01%	-0.02%	-0.03%	92%	95%

	Low delay B Main10				
	Over VTM-3.0				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	0.02%	-0.24%	0.07%	103%	108%
Class C	0.03%	0.21%	0.03%	100%	100%
Class E	0.16%	0.11%	-0.45%	107%	108%
Overall	0.06%	-0.01%	-0.07%	103%	105%
Class D	-0.01%	0.16%	0.00%	109%	108%
Class F (optional)	-0.02%	0.03%	0.56%	99%	93%

Spec Text (e.g. for Test 1)

In section 8.3.4.9 Derivation process for motion vector arrays from affine control point motion vectors, add

Text for Test 1

It is a requirement of bitstream conformance that

- If predFlagL0[xCb][yCb] is equal to 1 and predFlagL1[xCb][yCb] is equal to 1, $bxWX_4 * bxHX_4$ shall be less than or equal to $(256 \ll 22)$.
- Otherwise, if predFlagL0[xCb][yCb] is equal to 1 or predFlagL1[xCb][yCb] is equal to 1, $bxWX_h * bxHX_h$ shall be less than or equal to $(192 \ll 22)$.

Where the values of $(bxWX_4, bxHX_4)$ and $(bxWX_h, bxHX_h)$ are computed by

$$\begin{cases} maxW_4 = max(0,\ 4(2048 + \mathrm{dHorX}), 4\mathrm{dHorY}, 4(2048 + \mathrm{dHorX}) + 4\mathrm{dHorY}) \\ minW_4 = min(0,\ 4(2048 + \mathrm{dHorX}), 4\mathrm{dHorY}, 4(2048 + \mathrm{dHorX}) + 4\mathrm{dHorY}) \\ maxH_4 = \max\left(0,\ 4\mathrm{dVerX},\ 4(2048 + \mathrm{dVerY}), 4\mathrm{dVerX} + 4(2048 + \mathrm{dVerY})\right) \\ minH_4 = \min\left(0,\ 4\mathrm{dVerX},\ 4(2048 + \mathrm{dVerY}), 4\mathrm{dVerX} + 4(2048 + \mathrm{dVerY})\right) \end{cases}$$

$$\begin{cases} bxWX_4 = maxW_4 - minW_4 + (11 \ll 11) \\ bxHX_4 = maxH_4 - minH_4 + (11 \ll 11) \end{cases}$$

$$\begin{cases} bxWX_h = max\left(0,\ 4(2048 + \mathrm{dHorX})\right) - min\left(0,\ 4(2048 + \mathrm{dHorX})\right) + (11 \ll 11) \\ bxWX_h = max(0,\ 4\mathrm{dVerX}) - min(0,\ 4\mathrm{dVerX}) + (11 \ll 11) \end{cases}$$

Recommendation

- It is recommended to impose a bitstream restriction to constrain the memory bandwidth consumption of the affine mode at PU level.

Thanks to ByteDance for cross-check