JCTVC-T0059

On Adaptive motion vector resolution

Kai Zhang, Xianguo Zhang, Jicheng An, Han Huang,
Shawmin Lei

Overall Summary

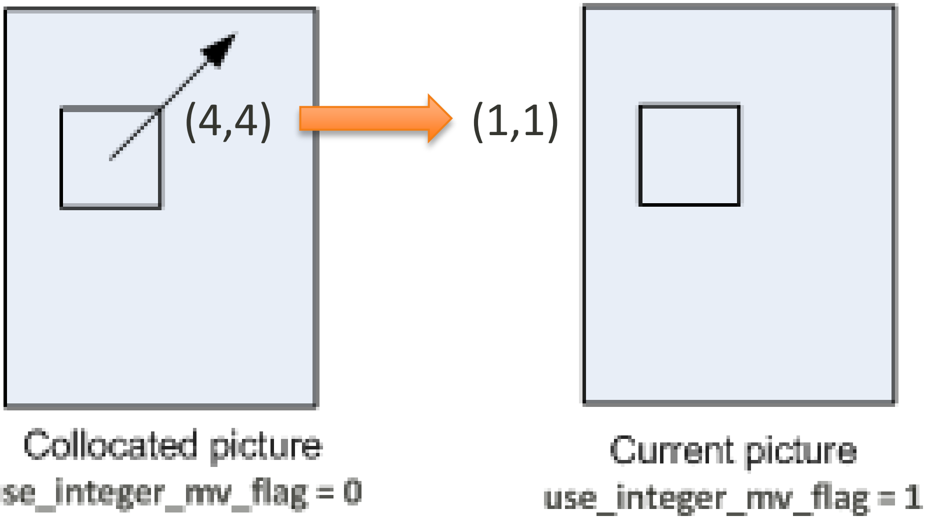
- Problems when adaptive motion vector resolution is applied
  - Mismatched MV resolution between TMVP and current MV
  - Unwanted larger MV range for motion compensation
- Proposed
  - Regularize the MV resolution of TMVP
  - Constrain the MV range when MV is in the integer-pixel resolution
- Results
  - Almost no performance change

Background

- Adaptive Motion resolution is adopted into SCM-3.0
  - A flag (**use_integer_mv_flag**) is signaled in slice header to indicate whether the MVs in the current slice are in integer-pixel or quarter-pixel resolution
  - When use_integer_mv_flag = 1
    - MVs are parsed and reconstructed are not changed
    - MV=MV<<2 before interpolation

Problems (1)

- TMVP and MV may have different MV resolution if **use_integer_mv_flag** in the current slice and in the col-pic are different
  - Example: If **use_integer_mv_flag** = 0 in col-pic but **use_integer_mv_flag** = 1 in the current slice. Then TMVP = (4, 4) should be (1,1) when it is considered in integer pixel resolution in current slice

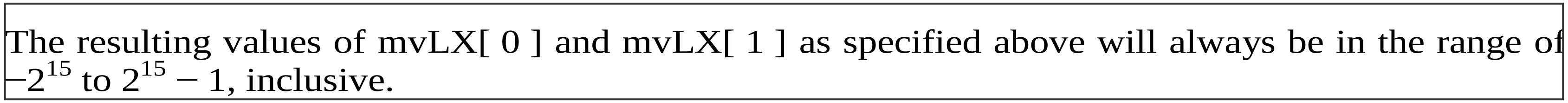


Proposed method-1

- When **use_integer_mv_flag** = 1
  - Store the MVs in temporal MV buffer in quarter-pixel precision
  - TMVP = TMVP >> 2 before it is used as MVP

Problems (2)

- MV range constraint may be violated
  - In the spec, it is constrained that

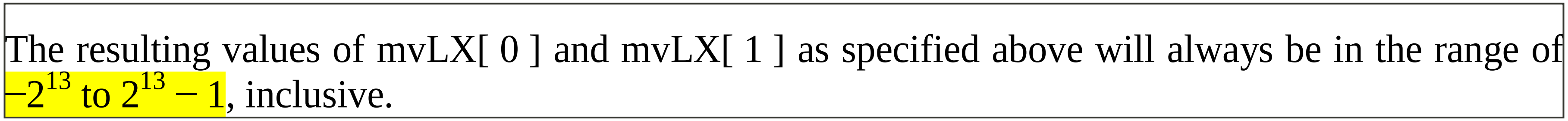


The resulting values of mvLX[0] and mvLX[1] as specified above will always be in the range of -2^{15} to $2^{15} - 1$, inclusive.

-
    - MV can always be represented and operated within 16 bits
    - The maximum accessing range (in integer-pixel precision) in the reference picture is within $[-2^{13}, 2^{13}-1]$
  - When **use_integer_mv_flag** = 1, MV is set to MV<<2 before interpolation
    - MV cannot be handled within 16 bits in interpolation
    - The maximum accessing range (in integer-pixel precision) in the reference picture becomes within $[-2^{15}, 2^{15}-1]$

Proposed method-2

- When **use_integer_mv_flag** = 1
  - – The constraint is modified as



The resulting values of mvLX[0] and mvLX[1] as specified above will always be in the range of -2^{13} to $2^{13} - 1$, inclusive.

Experiments

- Almost no performance change

	Random Access			Low delay B		
	G/Y	B/U	R/V	G/Y	B/U	R/V
RGB, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
RGB, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
RGB, Animation, 720p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
YUV, mixed content, 1440p & 1080p	0.0%	0.0%	0.0%	0.0%	-0.1%	-0.1%
YUV, Animation, 720p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]		100%			100%	
Dec Time[%]		99%			99%	

Thanks Microsoft for cross-checking

Conclusion

- Propose
  - Regularize MV resolution for TMVP when adaptive MV resolution is used
  - Constrain MV range depending on MV resolution

Thank you for your attention