

JVET-AG0149 Non-EE2: Improvements to Subblock Merge Mode

Jie Chen, Ru-ling Liao, Yeping Zheng, Xinwei Li, Yan Ye (Alibaba)

Introduction

- In ECM-11.0, template matching (TM) is applied on subblock merge mode.
 - For SbTMVP candidates: TM is applied to refine the motion shift which is used to locate the reference block in the collocated picture, the subblock MVs are not refined.
 - For affine candidates: TM is applied to refine the base MV of the affine model (i.e., all subblock MVs have a same MV offset), and it is only applied to uni-prediction candidates.
- In this contribution, it is proposed to improve the subblock merge mode by extending TM design and adding more SbTMVP candidates.
- Overall, it achieves $\{-0.22\%, -0.24\%, -0.26\%$ BD-rate gain in RA and $\{-0.19\%, 0.00\%, 0.04\%$ DB-rate gain in LD.

Proposed method

■ For SbTMVP candidate

- Add more candidates with all subblock reference indices set to 0 (max cand number increases from 5 to 7)
- Apply TM to refine the subblock MVs with a same MV offset
 - One top row and one left column which are fetched with the boundary subblock MVs are used as template.
 - TM search pattern is the same as current affine TM in ECM-11.0.

■ For affine candidates

- Extend TM process to bi-prediction candidates
 - The one with less initial TM cost between list 0 motion and list 1 motion is refined first and then the other one is refined.
 - For each motion refinement, the process is same with current design for uni-prediction.
- Add affine non-translation parameter refinement after base MV refinement
 - The refinement process is the same as that in affine DMVR with only BM being replaced by TM.
 - Only apply on uni-prediction candidates to reduce complexity.

$$\begin{cases} mv_x = a(x - x_0) + b(y - y_0) + base_mv_x \\ mv_y = c(x - x_0) + d(y - y_0) + base_mv_y \end{cases}$$

Simulation Results

- The proposed method was implemented on top of ECM-11.0 and tested under ECM CTC.
 - Overall, it gives 0.22% luma BD-rate reduction in RA and 0.19% luma BD-rate reduction in LD.
 - Higher coding gain is observed in 4K sequences.

	Random Access Main 10						
	Over ECM-11.0						
	Y	U	V	EncT	DecT	EncVmPeak	DecVmPeak
Class A1	-0.22%	-0.30%	-0.37%	103.9%	103.8%	100.2%	100.5%
Class A2	-0.33%	-0.27%	-0.35%	103.6%	105.6%	100.0%	100.1%
Class B	-0.16%	-0.21%	-0.19%	104.4%	104.5%	100.3%	100.6%
Class C	-0.22%	-0.21%	-0.19%	104.2%	106.3%	99.8%	100.9%
Class E							
Overall	-0.22%	-0.24%	-0.26%	104.1%	105.0%	100.1%	100.6%
Class D	-0.11%	-0.08%	-0.27%	103.4%	106.1%	100.8%	101.1%

	Low delay B Main 10						
	Over ECM-11.0						
	Y	U	V	EncT	DecT	EncVmPeak	DecVmPeak
Class A1							
Class A2							
Class B	-0.12%	0.33%	0.33%	102.5%	103.0%	100.4%	100.2%
Class C	-0.12%	0.30%	-0.23%	102.2%	103.8%	101.0%	100.7%
Class E	-0.41%	-0.96%	-0.08%	101.3%	102.8%	100.5%	100.6%
Overall	-0.19%	0.00%	0.04%	102.1%	103.2%	100.6%	100.5%
Class D	-0.32%	-1.03%	-1.58%	101.4%	103.2%	100.5%	101.0%

Conclusion

- It is proposed to extend the TM-based refinement in subblock merge mode and add more SbTMVP candidates to improve the coding efficiency.
- The experimental results show it achieves over 0.2% coding gain in RA with 4%/5% enc/dec run time increase.
 - Further code optimization is expected to reduce the run time increase.
- More coding gain is observed in 4K sequences.
- It is suggested to study the proposed method in the next round of EE2.

Thanks

