Decoder-side Affine Model Refinement for Video Coding beyond VVC

Jie Chen, Ru-ling Liao, Yan Ye and Xinwei Li

March 24 , 2023

Background

- Versatile Video Coding (VVC) was finalized on July 2020
  - Developed by Joint Video Experts Team (JVET)
  - Achieves about half bit-rate reduction compared with its predecessor high efficiency video coding (HEVC) [2]
- New video coding technology exploration beyond VVC was launched by JVET on 2021
  - The exploration platform, Enhanced Compression Model (ECM)-1.0, was released in April 2021
  - By December 2022, ECM-7.0 achieves 19.2% luma BD rate savings over VTM-11.0
  - Quite a lot of technology improvement and new coding tools were adopted in ECM
- It is proposed to refine the affine model for affine motion compensation blocks on top of ECM software in this paper

Affine Motion Model (1/2)

- The affine motion can be described by a affine motion model with 6 or 4 parameters
  - 6 parameter affine model
  - 4 parameter affine model

- The 6-parameter affine model is a generalized linear mapping which can preserve lines and parallelism.

$$\begin{cases} mv_x = a(x - x_0) + c(y - y_0) + mv_{0x} \\ mv_y = b(x - x_0) + d(y - y_0) + mv_{0y} \end{cases} \quad (1)$$

- The 4-parameter affine model is a subset of 6-parameter affine model but still can support translation, rotation and zooming.

$$\begin{cases} mv_x = a(x - x_0) - b(y - y_0) + mv_{0x} \\ mv_y = b(x - x_0) + a(y - y_0) + mv_{0y} \end{cases} \quad (2)$$

(mv_x, mv_y): MV at any coordinate (x, y) in the plane
(mv_{0x}, mv_{0y}): base MV controlling the translation movement of the model
a, b, c and d: non-translation parameters defining rotation, zooming, shearing and other more complex motion

Affine Motion Model (2/2)

- The affine motion model can be determined by the control point motion vectors (CPMVs)
  - CPMVs of a affine coded blocks are stored in VVC and ECM
  - 3 CPMVs for 6 parameter affine model

$$\begin{cases} a = \frac{mv_{1x} - mv_{0x}}{x_1 - x_0} \\ b = \frac{mv_{1y} - mv_{0y}}{x_1 - x_0} \\ c = \frac{mv_{2x} - mv_{0x}}{y_2 - y_0} \\ d = \frac{mv_{2y} - mv_{0y}}{y_2 - y_0} \end{cases} \quad (3)$$

  - 2 CPMVs for 4 parameter affine model

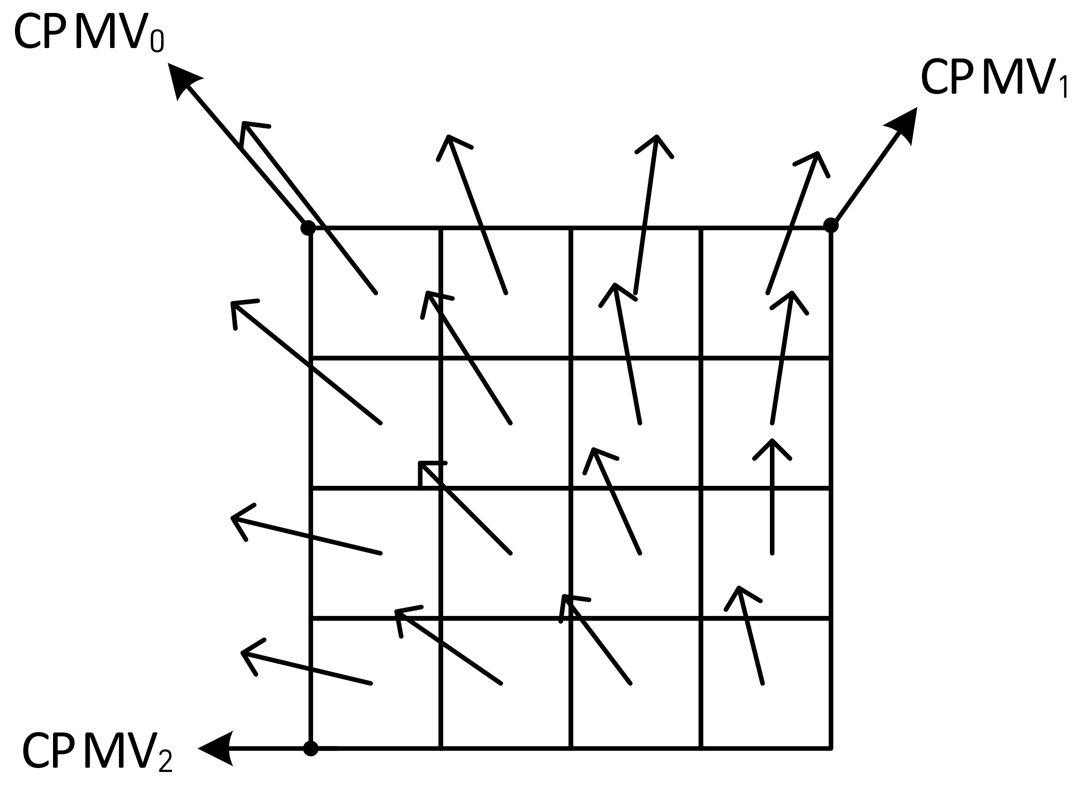
$$\begin{cases} a = \frac{mv_{1x} - mv_{0x}}{x_1 - x_0} \\ b = \frac{mv_{1y} - mv_{0y}}{x_1 - x_0} \end{cases} \quad (4)$$

(mv_{0x}, mv_{0y}), (mv_{1x}, mv_{1y}) and (mv_{2x}, mv_{2y}): CPMVs which are MVs at top-left, top-right and bottom-left corner of CU
(x_0, y_0), (x_1, y_1) and (x_2, y_2): coordinates of top-left, top-right and bottom-left corner of CU

Affine motion compensation (AMC)

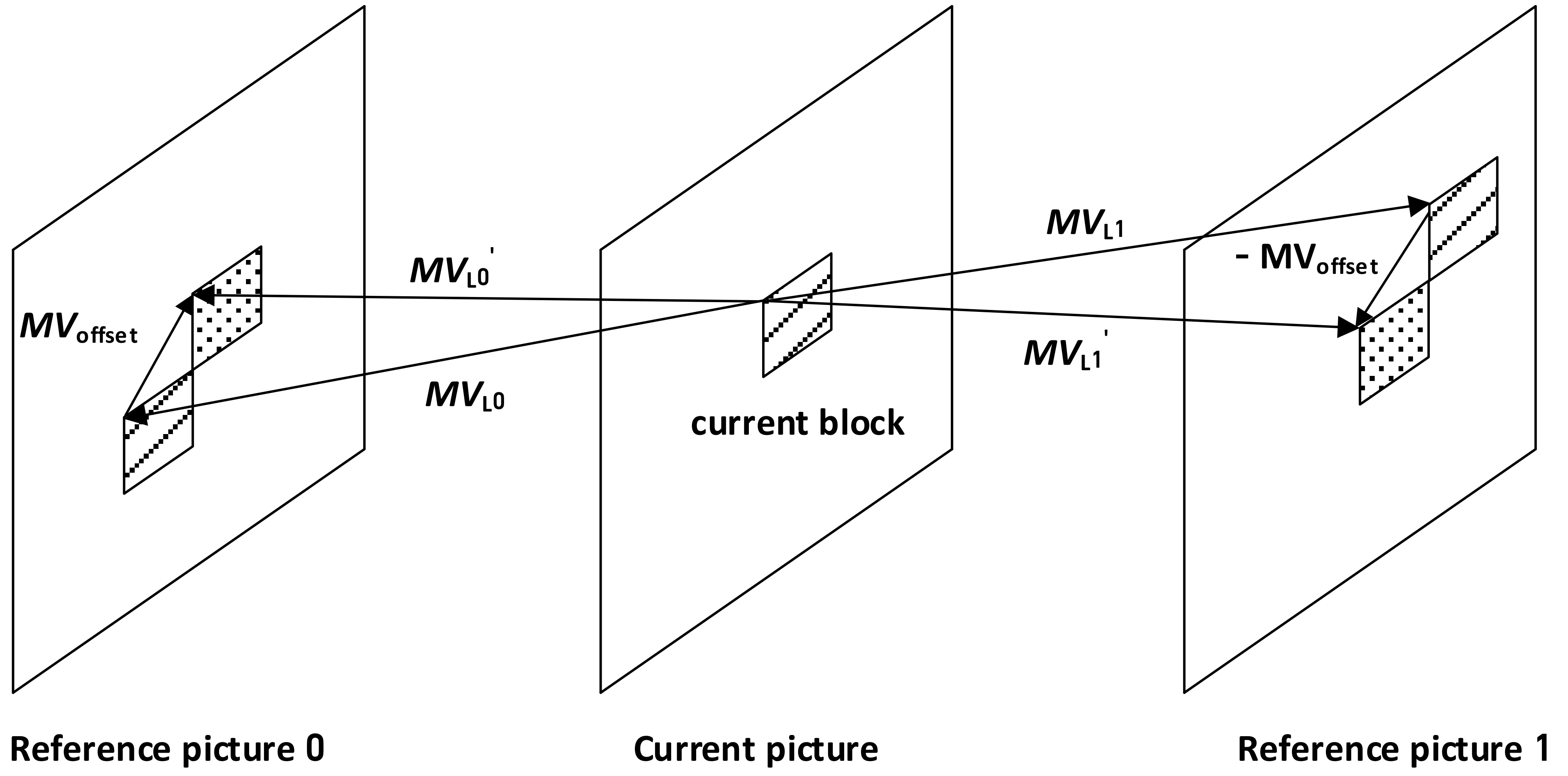
- Two affine motion compensation modes, affine inter mode and affine merge mode, are supported in VVC and ECM
  - ➢ Affine inter mode: an inter mode requiring CPMV differences signaled (2 differences for 4-parameter affine model and 3 differences for 6-parameter affine model). CPMVs are derived by adding CPMV differences to CPMV predictors.
  - ➢ Affine merge mode: a merge mode where motion info is derived by the neighboring bocks. The CPMVs are derived from a affine merge list. The index to the affine merge list is signaled.

- Subblock level motion compensation is performed for AMC instead of pixel level
  - Subblock MVs are derived based on the CPMVs
  - Subblock motion compensation is performed with subblock MV



Decoder-side Motion Vector Refinement (DMVR)

- DMVR was adopted in VVC and further extended in ECM which aims at increasing the motion vector accuracy without additional signaling
  - ➢ Applied on merge mode and based on a search scheme in both decoder and encoder
  - ➢ Applied on bi-prediction block as it is based on bilateral matching (BM) cost (e.g., the SAD/SATD between two predictors)
  - ➢ initial MVs: the MV pair derived in merge mode
  - ➢ refined MVs: The MV pair with the minimum BM cost in the search window around initial MVs



The method proposed

- In VVC and ECM-5.0, DMVR can only be applied on translation motion compensated blocks
- For affine merge mode, the affine motion is inherited or derived from the previously coded blocks and may not well match with the current block
- It is proposed to refine the affine motion model in decoder side for AMC blocks to improve the affine model accuracy
    - ➢ Similar with conventional DMVR
        - ✓ Applied on affine merge and based on a search scheme both in encoder and decoder
        - ✓ Applied on bi-predicted block with one reference forward and on reference backward
        - ✓ No additional signaling – always on if condition satisfied.
    - ➢ It consists of two steps
        - ✓ Base MV refinement
        - ✓ Non-translation parameter refinement

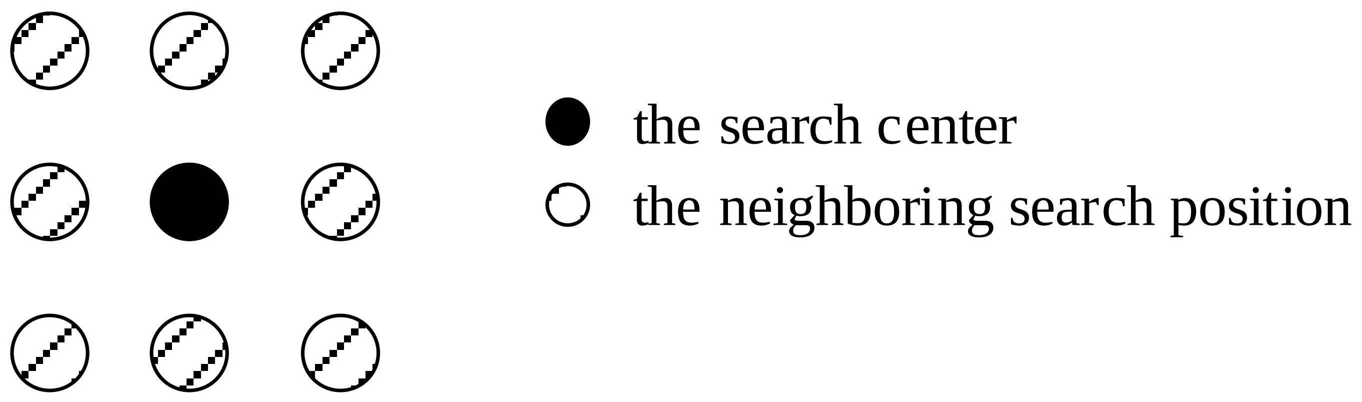
Base MV refinement

- Only base MV of the affine model is refined and the non-translation parameters are unchanged
  - ➢ The MV offset for base MV is searched to minimize the BM cost
  - ➢ The MV offset search process also obey the symmetrical rule

$$\begin{cases} mv'_{0x\_L0} = mv_{0x\_L0} + mv_{offsetx} \\ mv'_{0y\_L0} = mv_{0y\_L0} + mv_{offsety} \\ mv'_{0x\_L1} = mv_{0x\_L1} - mv_{offsetx} \\ mv'_{0y\_L1} = mv_{0y\_L1} - mv_{offsety} \end{cases} \quad (5)$$

(mv_{0x_L0}, mv_{0y_L0}), (mv_{0x_L1}, mv_{0y_L1}): initial base MV pair
$(mv'_{0x\_L0}, mv'_{0y\_L0})$, $(mv'_{0x\_L1}, mv'_{0y\_L1})$: refined base MV pair
$(mv_{offsetx}, mv_{offsety})$: the MV offset obtained in the search process

  - ➢ A search process followed by a parametric error surface based fractional MV estimation
    - ✓ Integer search followed by half pixel search
    - ✓ 3x3 square search pattern is used



Non-translation parameter refinement

- The refined base MV is kept unchanged and non-translation parameters are refined
  - The parameter offsets are searched to minimize the BM cost
  - The parameter offset search process also obey the symmetrical rule

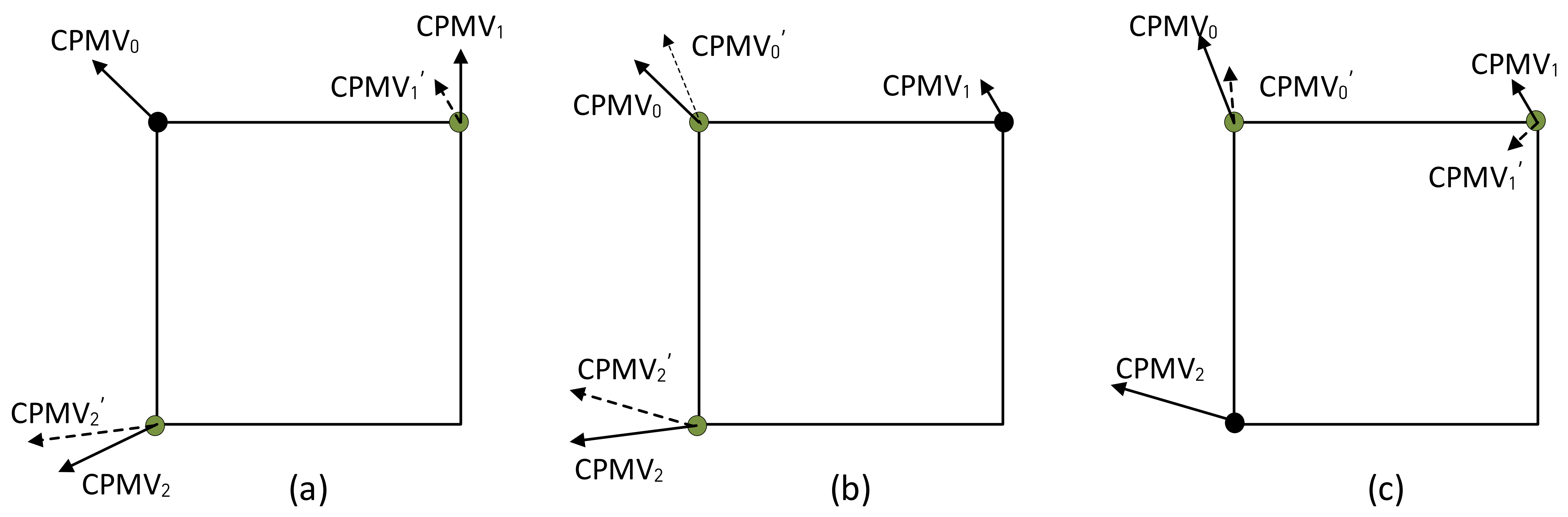
$$\begin{cases} a'_{L0} = a_{L0} + a_{offset} \\ a'_{L1} = a_{L1} - a_{offset} \\ b'_{L0} = b_{L0} + b_{offset} \\ b'_{L1} = b_{L1} - b_{offset} \\ c'_{L0} = c_{L0} + c_{offset} \\ c'_{L1} = c_{L1} - c_{offset} \\ d'_{L0} = d_{L0} + d_{offset} \\ d'_{L1} = d_{L1} - d_{offset} \end{cases} \quad (6)$$

$a_{L0}/\ a_{L1}$, b_{L0}/b_{L1}, c_{L0}/c_{L1} and d_{L0}/d_{L1}: initial non-translation parameters
a'_{L0}/a'_{L1}, b'_{L0}/b'_{L1}, c'_{L0}/c'_{L1} and d'_{L0}/d'_{L1}: refined non-translation parameters
a_{offset}, b_{offset}, c_{offset} and d_{offset}: parameter offsets obtained in the search process

  - Different search patterns for different affine motion models
    - ✓ Square search pattern is used for 6-parameter affine motion model
      - (a_0, b_0, c_0, d_0) → $(a_0 - s, b_0, c_0, d_0)$, $(a_0, b_0 - s, c_0, d_0)$, $(a_0, b_0, c_0 - s, d_0)$, $(a_0, b_0, c_0, d_0 - s)$, $(a_0 + s, b_0, c_0, d_0)$, $(a_0, b_0 + s, c_0, d_0)$, $(a_0, b_0, c_0 + s, d_0)$, $(a_0, b_0, c_0, d_0 + s)$
    - ✓ Cross search pattern is used for 4-paraterm affine motion model
      - (a_0, b_0) → $(a_0 - s, b_0 - s)$, $(a_0 - s, b_0)$, $(a_0 - s, b_0 + s)$, $(a_0, b_0 + s)$, $(a_0 + s, b_0 + s)$, $(a_0 + s, b_0)$, $(a_0 + s, b_0 - s)$, $(a_0, b_0 - s)$

Non-translation parameter refinement

- Iteration process is introduced for non-translation parameter refinement
  - Iteration 1: the top-left CPMV is fixed as base MV, two other CPMVs are refined
  - Iteration 2: the refined top-right CPMV is fixed as base MV, two other CPMV are further refined
  - Iteration 3: the refined bottom-left CPMV is fixed as base MV, two other CPMVs are further refined



Experimental Results

- To show the effectiveness of the proposed method, the percentage of area of the blocks coded with affine merge mode is collected

Table 1: The comparison of the area percentage of the blocks coded with affine merge mode

Sequence	ECM-6.0 anchor	Base MV refinement	Both base MV and non-translation parameter refinement
Tango2	5.22%	8.99%	12.97%
FoodMarket4	13.38%	18.08%	20.86%
Campfire	2.76%	3.09%	3.38%
CatRobot1	5.89%	8.28%	10.51%
DaylightRoad2	10.42%	15.50%	19.39%
ParkRunning3	20.00%	25.75%	27.73%
MarketPlace	10.83%	14.63%	16.99%
RitualDance	8.19%	10.56%	12.25%
Cactus	5.49%	6.51%	7.27%
BasketballDrive	5.70%	7.09%	8.14%
BQTerrace	3.51%	3.53%	3.64%
BasketballDrill	2.35%	2.72%	3.60%
BQMall	1.87%	2.39%	2.92%
PartyScene	6.80%	7.70%	7.73%
RaceHorses	4.17%	5.93%	8.19%
Average	**7.11%**	**9.38%**	**11.04%**

Experimental Results

- BD-rate reduction of the proposed method on top of ECM-6.0

Table 2: Simulation results of the proposed method (Random Access)

Class	Sequence	Base MV refinement			Both base MV and non-translation parameter refinement		
		Y	U	V	Y	U	V
A1	Tango2	-0.23%	-0.13%	-0.21%	-0.64%	-0.55%	-0.49%
	FoodMarket4	-0.33%	-0.40%	-0.32%	-0.49%	-0.47%	-0.24%
	Campfire	-0.02%	-0.01%	0.03%	-0.04%	-0.03%	-0.03%
A2	CatRobot1	-0.26%	-0.09%	-0.18%	-0.47%	-0.36%	-0.44%
	DaylightRoad2	-0.48%	-0.41%	-0.29%	-0.86%	-0.53%	-0.57%
	ParkRunning3	-0.15%	-0.07%	-0.09%	-0.19%	-0.11%	-0.08%
B	MarketPlace	-0.23%	-0.12%	-0.20%	-0.41%	-0.27%	-0.40%
	RitualDance	-0.10%	0.12%	-0.16%	-0.21%	0.13%	-0.20%
	Cactus	-0.15%	0.00%	-0.06%	-0.20%	-0.20%	-0.08%
	BasketballDrive	-0.06%	0.09%	-0.07%	-0.17%	-0.04%	-0.13%
	BQTerrace	-0.02%	0.18%	0.11%	-0.01%	0.23%	-0.08%
C	BasketballDrill	-0.05%	0.05%	-0.20%	-0.10%	0.03%	-0.18%
	BQMall	-0.05%	0.02%	-0.01%	-0.10%	0.13%	-0.02%
	PartyScene	-0.03%	-0.01%	0.20%	-0.03%	-0.05%	0.28%
	RaceHorses	-0.01%	-0.12%	-0.25%	-0.09%	-0.24%	-0.14%
Per-Class	Class A1	-0.20%	-0.18%	-0.17%	-0.39%	-0.35%	-0.25%
	Class A2	-0.30%	-0.19%	-0.19%	-0.50%	-0.33%	-0.36%
	Class B	-0.11%	0.06%	-0.08%	-0.20%	-0.03%	-0.18%
	Class C	-0.04%	-0.01%	-0.06%	-0.08%	-0.03%	-0.01%
Overall		**-0.15%**	**-0.06%**	**-0.11%**	**-0.27%**	**-0.15%**	**-0.19%**
EncT			102%			113%	
DecT			102%			113%	

Experimental Results

- Further work has been done to reduce the complexity of the non-translation parameter refinement
  - Anchor: ECM-7.0 (with base MV refinement)
  - Test: non-translation parameter refinement

Table 3: Simulation results on ECM-7.0

Class	Sequence	Non-translation parameter refinement		
		Y	U	V
A1	Tango2	-0.23%	0.00%	0.05%
	FoodMarket4	-0.20%	-0.27%	-0.27%
	Campfire	-0.02%	-0.01%	0.09%
A2	CatRobot1	-0.20%	-0.02%	-0.03%
	DaylightRoad2	-0.22%	-0.24%	-0.40%
	ParkRunning3	-0.05%	0.01%	-0.07%
B	MarketPlace	-0.04%	0.14%	0.01%
	RitualDance	-0.03%	-0.07%	-0.14%
	Cactus	-0.16%	-0.16%	-0.06%
	BasketballDrive	-0.07%	-0.06%	-0.19%
	BQTerrace	-0.05%	-0.11%	-0.19%
C	BasketballDrill	-0.04%	-0.12%	0.04%
	BQMall	-0.03%	0.17%	0.08%
	PartyScene	0.05%	-0.13%	0.00%
	RaceHorses	-0.13%	-0.05%	-0.29%
Per-Class	Class A1	-0.15%	-0.09%	-0.04%
	Class A2	-0.16%	-0.08%	-0.17%
	Class B	-0.07%	-0.05%	-0.11%
	Class C	-0.04%	-0.03%	-0.04%
Overall		-0.10%	-0.06%	-0.09%
	EncT		103%	
	DecT		103%	

Summary

- It is proposed to refine affine motion model for affine merge coded blocks to increase the model accuracy

- Two steps are in the proposed method
  - Base MV refinement
  - Non-translation parameter refinement

- The experimental results show the BD-rate reduction on top of ECM
  - {0.15% (Y), 0.06% (U), 0.11% (V), 102%(ENC), 102%(DEC)} on ECM-6.0, if only base MV refinement is applied
  - {0.27% (Y), 0.15% (U), 0.19% (V), 113%(ENC), 113%(DEC)} on ECM-6.0 if both two steps are applied
  - {0.10% (Y), 0.06% (U), 0.09% (V), 103%(ENC), 103%(DEC)} on ECM-7.0 when non-transaltion paramter refinement is applied
  - More coding gain on 4K sequences

Reference

1. B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, J. R. Ohm, “Overview of the versatile video coding (VVC) standard and its applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, issue 10, pp 3736-3764, Aug. 2021.
2. G. J. Sullivan, J. Ohm, W. Han, T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
3. ECM-1.0, https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-1.0
4. VTM software, https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM
5. ECM-6.0, https://vcgit.hhi.fraunhofer.de/ecm/ECM/-/tree/ECM-6.0
6. S. Esenlik and Y.-W. Chen, “CE9: Summary report on decoder side MV derivation,” JVET-K0029, Ljubljana, SI, Jul. 2018.
7. S. Esenlik, Y.-W. Chen, and F. Chen, “CE9: Summary report on decoder side motion vector derivation,” JVET-L0029, Macao, CN, Oct. 2018.
8. X. Xiu and S. Esenlik, “CE9: Summary report on decoder side motion vector derivation”, JVET-M0029, Marrakech, MA, Jan. 2019.
9. H. Gao, X. Chen, S. Esenlik, J. Chen and E. Steinbach, “Decoder-side motion vector refinement in VVC: algorithm and hardware implementation considerations,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3197-3211, Aug. 2021.
10. M. Coban, F. Le Léannec, K. Naser, J. Ström, “Algorithm description of enhanced compression model 6 (ECM 6),” JVET-AA2025, Jul. 2022.
11. H. Huang, J. Woods, Y. Zhao, and H. Bai, “Control-point representation and differential coding affine-motion compensation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp.1651–1660, Oct. 2013.
12. N. Zhang, X. Fan, D. Zhao, and W. Gao, “Merge mode for deformable block motion information derivation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2437-2449, Nov. 2017.
13. L. Li, H. Li, D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, and F. Wu, “An efficient four-parameter affine motion model for video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1934-1948, Apr. 2017.
14. K. Zhang, Y. Chen, L. Zhang, W. Chien and M. Karczewicz, “An improved framework of affine motion compensation in video coding,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1456-1469, Mar. 2019.
15. H. Yang, H. Chen, J. Chen, S. Esenlik, S. Sethuranman, X. Xiu, E. Alshina and J. Luo, “Subblock-based motion derivation and inter prediction refinement in the versatile video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3862-3877, Oct. 2021.
16. M. Karczewice, Y. Yan, “Common test conditions and evaluation procedures for enhanced compression tool testing,” JVET-Y2017, teleconference, Jan. 2022.
17. G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” ITU-T SG16 Q.6 Document, VCEG-M33, Austin, US, Apr. 2001.

Thank you