AFFINE DIRECT/SKIP MODE WITH MOTION VECTOR DIFFERENCES IN VIDEO CODING

Tianliang Fu[1], Kai Zhang[2], Hongbin Liu[3], Li Zhang[2], Shanshe Wang[4], Siwei Ma[4], Wen Gao[4]

[1]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School
[2]Bytedance Inc., San Diego CA. 92122 USA.
[3]Beijing Bytedance Network Technology Co., Ltd 56 Zhichun Rd, Haidian District Beijing, China
[4]Institute of Digital Media, Peking University, Beijing, China
{futl, sswang, swma, wgao}@pku.edu.cn, {zhangkai.video, liuhongbin.01, lizhang.idm}@bytedance.com

ABSTRACT

Audio and Video coding Standard-3 (AVS-3) is an emerging next-generation video coding standard beyond AVS-2. New technologies, such as affine motion compensation and ultimate motion vector expression have been adopted in AVS-3 phase2. In this paper, Affine Direct/Skip mode with Motion Vector Differences (ADS-MVD) is proposed to improve the existing affine direct/skip mode in AVS-3 phase2. With ADS-MVD, control point motion vectors of a chosen direct/skip merge candidate are refined by adding offsets. The offsets are selected from predefined offset candidates and the indices of the selected offsets are signaled. Simulation results demonstrate that the proposed method can achieve 0.20% BD-rate savings for Y components on average under the random-access configurations with a negligible encoding time/decoding time change. On sequences with rich affine motions, the proposed method achieves a coding gain of up to 0.45%. The method of ADS-MVD has been adopted into AVS-3 phase2.

Index Terms— Affine, Direct/Skip Mode, AVS-3

1. INTRODUCTION

Video compression technologies play a critical role in the emerging mobile internet era which has tremendous demand for high-quality video content. With the video data exploding dramatically, there is an ever-increasing demand for the development of video compression technologies which would provide a significantly higher coding efficiency than current video coding standards. Targeting at an even higher compression efficiency than AVS [1], experts in the AVS working group of China collaborated to develop a new generation video coding standard, named AVS-2 [2] in 2016. It is demonstrated that AVS-2 can reduce roughly 50% bitrate while achieving approximately the same subjective quality as AVS. Although AVS-2 improve the coding performance of AVS dramatically, a more efficient video compression scheme is desired with the sustained and rapid growth of bandwidth demand for video content.

AVS-3 is an emerging next-generation video coding standard beyond AVS-2. In the first phase of AVS-3 standardization, numerous efficient tools have been studied and adopted. As an infrastructure design, AVS-3 adopts a partitioning structure comprising quad-tree/binary-tree (QTBT) [3] plus extend quad-tree (EQT) [4], to achieve a partitioning pattern much more flexible than that in AVS-2. In terms of intra-frame prediction, intra prediction filter (IPF) [5] and two-step cross-component prediction mode (TSCPM) [6] have been adopted to generate the intra-prediction with higher quality. Regarding inter-frame prediction, affine motion compensation [7], ultimate motion vector expression (UMVE) [8], history-based motion vector prediction (HMVP) [9] and advanced motion vector resolution (AMVR) [10] have been adopted to capture various motion characteristics, leading to a great improvement of inter coding efficiency.

Affine motion model (AMM) has been studied for decades because it can describe complex motions such as zooming, rotation or shearing efficiently. In recent years, significant progress has been made in balancing the precision of AMM and the bits spent on representing AMM. Huang [11] proposed a 6-parameter AMM represented by three Control Point Motion Vectors (CPMVs). Zhang [12] inserted AMM direct/skip candidates into a direct/skip candidate list. Li [13] further developed the CPMV-based AMM representation method and proposed a 4-parameter AMM represented by two CPMVs. An iterative gradient-based fast ME method is devised for low-complexity encoders. Meanwhile, the MV granularity in the AMM is changed from pixel level to 4×4 sub-block level, to restrain the decoder complexity. The affine motion compensation (AMC)

978-1-7281-1485-9/20/$31.00 ©2020 IEEE

framework was adopted into VVC in July 2018 with further improvements [14]. Although CPMV-based AMM can achieve a significant coding gain in AVS-3, it still requires more bits on coding MVD of CPMVs as compared to translation motion model (TMM).

UMVE provides a simplified signaling method to express MVs. The expression method includes a base MV, a motion magnitude and a motion direction. UMVE can achieve a better trade-off between the inter-mode and the direct/skip mode. Compared with the inter mode, the overhead bits of MVD in UMVE is less. Compared with the direct/skip mode, the MV signaled by UMVE is more accurate.

To further explore the potential advantage of AMC and UMVE, this paper proposes a method of Affine Direct/Skip mode with Motion Vector Differences (ADS-MVD), extending the concept of UMVE to affine direct/skip mode. With ADS-MVD, each CPMV of an affine direct/skip candidate can be refined with one of the predefined candidate control point motion vector differences (CPMVDs). The candidate CPMVDs are generated by combining five predefined motion vector magnitudes and four predefined motion vector directions.

The rest of this paper is organized as follows: Section 2 reviews affine motion compensation and UMVE for TMM in AVS-3. Section 3 describes the proposed ADS-MVD. Simulation results are reported in Section 4 and conclusions are drawn in Section 5.

2. BACKGROUND

2.1. Affine Motion Compensation in AVS-3

In AVS-2, only TMM is applied in motion compensation. While in the real world, there are varieties of motions, e.g. zoom in/out, rotation, perspective motions and other non-translational motions. In order to address these non-translational motions, a block-based affine motion compensation prediction is applied in AVS-3. As shown in Fig.1, the affine motion field of a block is represented by motion information at two control points with the 4-parameter affine motion model. With the 4-parameter affine motion model, motion vector at sample position (x, y) in a block is derived as:

$$\begin{cases} MV_x = \dfrac{mv_1^x - mv_0^x}{w} x - \dfrac{mv_1^y - mv_0^y}{w} y + mv_0^x \\ MV_y = \dfrac{mv_1^y - mv_0^y}{w} x + \dfrac{mv_1^x - mv_0^x}{w} y + mv_0^y \end{cases} \quad (1)$$

Where (mv_0^x, mv_0^y) is CPMV at the top-left corner, (mv_1^x, mv_1^y) is CPMV at the top-right corner and w is the width of current CU.

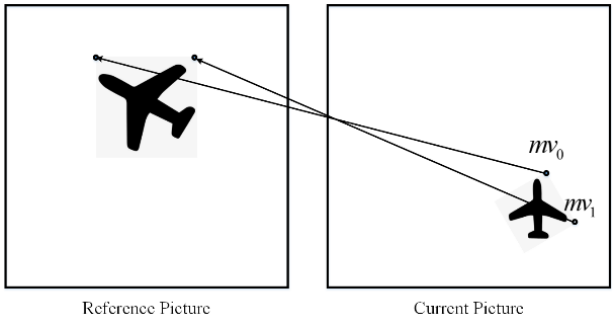


Fig.1. 4-parameter affine model.

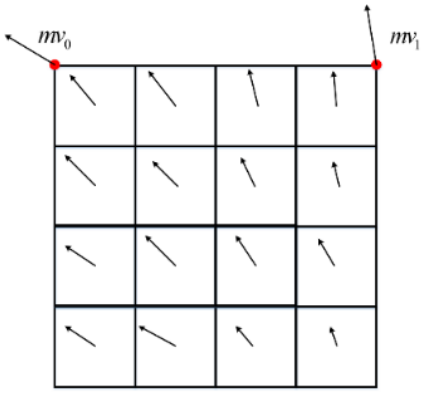
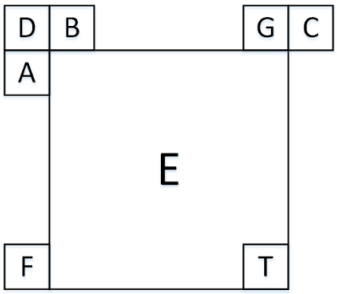


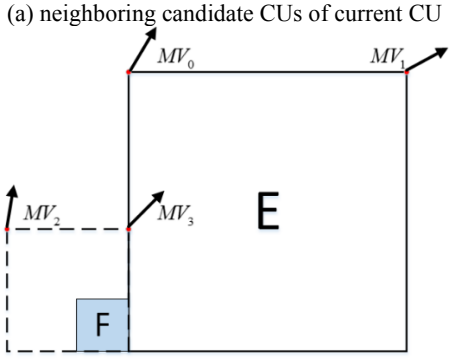
Fig.2. Subblock-based AMC.

To reduce the complexity of AMC, the MV granularity in AVS-3 is at the sub-block level instead of the pixel level. As shown in Fig.2, in order to derive MV of each sub-block, the MV at the center point is calculated according to Eq (1). All samples within the sub-block share the MV at the center point. AMC is applied to generate the prediction samples of each sub-block with derived MVs. In AVS-3, the size of the sub-block is set to be 4x4 for unidirectional prediction and 8x8 for bidirectional prediction. Similar to the translation motion model, there are two affine motion prediction modes: affine AMVP mode and affine direct/skip mode.

With the affine AMVP mode, CPMVs of a CU can be predicted from MVs of neighboring CUs. Block A, B, D in Fig.3(a) are checked to derive $CPMV_0$ and block G, C are checked to derive $CPMV_1$. Different from affine direct/skip mode, CPMVDs need to be signaled in the bitstream. In order to find the best CPMV of current CU, gradient-based affine ME is performed in HPM-5.0, which can solve the two CPMVs simultaneously at each iteration and converge to the optimal combination quickly.



(a) neighboring candidate CUs of current CU



(b) constructed affine direct/skip candidates

Fig. 3. Neighboring blocks used in AMC.

With the affine direct/skip mode, the CPMVs of the current CU is generated based on the motion information of spatial and temporal neighboring CUs. An affine direct/skip candidate list needs to be constructed first and the size of candidate list is five. To construct the candidate list, three types of affine direct/skip candidates are involved: inherited candidates, constructed candidates and zero candidates. There are up to two inherited candidates in AVS-3, which are derived by the affine motion model with neighboring affine-coded CUs. The neighboring candidate CUs are shown in Fig.3 (a) and the scanning order is F, G, C, A, D. When a neighboring affine-coded CU is selected, its control point motion vectors are used to derive the CPMVs of the current CU. For example, if block F is coded with affine mode, the two CPMVs of current CU are calculated according to the AMC, with MVs at the top-left corner and top-right corner of the neighboring CU, *i.e.* mv_2 and mv_3 as shown in Fig.3 (b). Constructed candidates are derived by combining the neighboring motion information of four corner control points. First, four CPMVs are generated by checking the specified spatial neighbors and temporal neighboring blocks. The checking order for $CPMV_0$ is A, B and D; for $CPMV_1$ is G and C; for $CPMV_2$ is block F and for $CPMV_3$ is the temporal neighboring block T. After generating MVs of four corner control points, affine direct/skip candidates are constructed based on different combinations of the CPMVs. After checking inherited candidates and constructed candidate, if the candidate list is still not full, zero MVs are inserted to the end of the list.

2.2. UMVE in AVS-3

With the direct/skip mode in AVS-3, the implicitly derived motion information is directly used for MC of the current CU. To predict the current CU more accurately, a direct/skip candidate can be further refined by a signaled MVD when UMVE is applied. Different from inter AMVP mode, only several predefined MVDs around zero MVD is allowed in UMVE. In addition, an index to indicate the MVD magnitude and an index to indicate the MVD direction are signaled to represent the MVD instead of signaling MVD directly.

Distance index indicates the magnitude of the MVD, *i.e.* an offset away from the candidate MV. The offset is added to either the horizontal component or the vertical component of candidate MV according to a direction index. To get a good trade-off between performance and complexity, five distance indices are predefined in UMVE: 1/4-pixel, 1/2-pixel, 1-pixel, 2-pixel and 4-pixel. A direction index represents the direction of the MVD and only four directions are allowed: the positive direction along the x-axis, the negative direction along the x-axis, the positive direction along the y-axis and the negative direction along the y-axis. Taking both distance index and direction index into consideration, the allowed predefined MVDs are shown in Fig.4. Table 1/Table 2 reveals the mapping between indices and MVD offsets/directions.

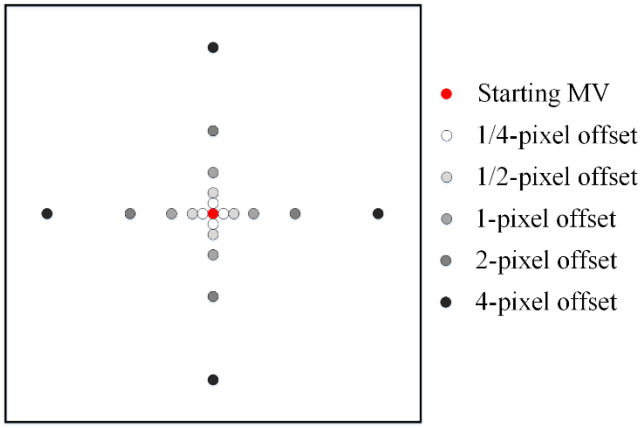


Fig.4. UMVE allowed search point

Table1. The mapping between distance indices and offsets

Distance Index	0	1	2	3	4
Offset (in pixel)	1/4	1/2	1	2	4

Table2. The mapping between direction indices and directions

Direction Index	00	01	10	11
x-axis	+	-	N/A	N/A
y-axis	N/A	N/A	+	-

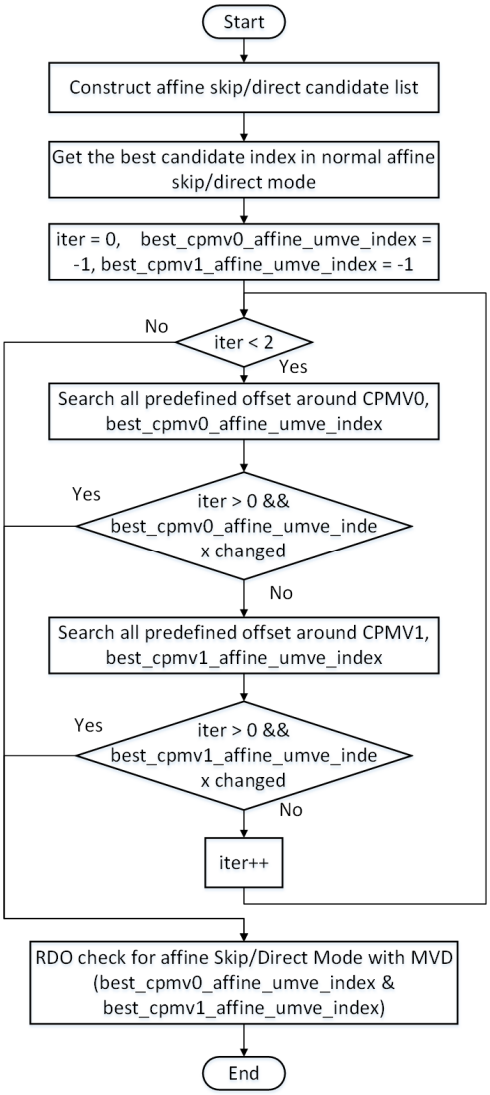


Fig.5. A flowchart of ADS-MVD at the encoder.

3. AFFINE DIRECT/SKIP MODE WITH MOTION VECTOR DIFFERENCES

The affine AMVP mode can provide AMM with higher precision than affine direct/skip mode by signaling CPMVDs directly with a penalty of overhead bits. To achieve a better trade-off between the precision of AMM and the overhead bits, ADS-MVD extends the concept of UMVE to AMM and it has been adopted into AVS-3 video coding standard.

With ADS-MVD, the affine direct/skip candidate list is constructed in the same way as the normal affine direct/skip mode where are up to five candidates in the candidate list. After a candidate is selected, the CPMVs of the candidate are further refined by adding CPMVDs. The CPMVDs are signaled in a way like UMVE, including a distance index and a direction index for each control point. Same to the normal UMVE, five types of distance (1/4-pixel, 1/2-pixel, 1-pixel, 2-pixel and 4-pixel) and four directions (the positive direction along the x-axis, the negative direction along the x-axis, the positive direction along the y-axis and the negative direction along the y-axis) are allowed for each control point. The mapping between indices and CPMVD offsets/directions are also shown in Table 1/Table 2.

In AVS-3, two control points are used to represent the affine model. In the proposed ADS-MVD, there are 20 predefined CPMVDs (5 distance × 4 directions) for each control point. A straight-forward way is to search all the combinations of $CPMVD_0$ and $CPMVD_1$. However, such a search method will lead to huge coding complexity since 400 (20×20) combinations of $CPMVD_0$ and $CPMVD_1$ need to be checked for one candidate. Since the Rate-Distortion Optimization (RDO) check is applied for each combination, the full-search method is infeasible at the encoder. To accelerate the encoder, three fast strategies are proposed.

First, SATD is used to roughly estimate the cost when searching for best combination. The combination of $CPMVD_0$ and $CPMVD_1$ with the smallest SATD cost is chosen to perform the RDO check in the next step, to compare with other modes according to the RDO criterion. The ADS-MVD mode is selected when the RD cost of the selected combination is less than those of all other modes.

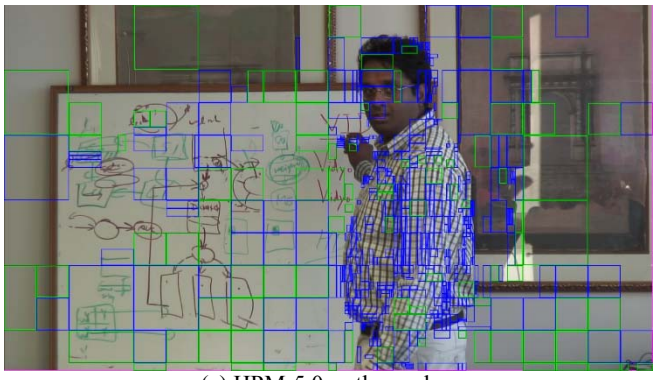
Second, the normal affine direct/skip mode is performed before the ADS-MVD mode. The best affine candidate selected by the normal affine direct/skip mode can be directly utilized as the best candidate in the ADS-MVD mode, which can reduce the number of combinations greatly while coding performance drops only a little.

Third, it is proposed to optimize one CPMVD out of the two CPMVDs iteratively to further reduce the checking number of combinations for each candidate. Specially, all predefined CPMVDs applied on $CPMV_0$ are checked as $CPMV_1$ fixed. After finding the best $CPMVD_0$ for $CPMV_0$, all predefined CPMVDs around $CPMV_1$ are checked as $CPMV_0$ added by the best $CPMVD_0$ is fixed. The process can be repeated iteratively. During the iterative process, an early termination method is implemented to remove redundant checking. When the iteration is not the first iteration and the best CPMVD of current checking CPMV does not change, the process can be early terminated.

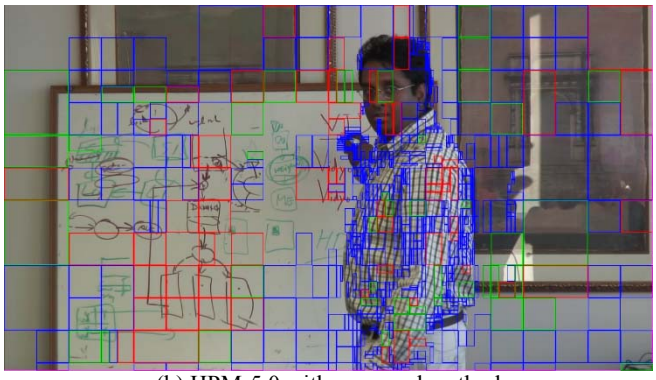
As a conclusion, a flowchart of the algorithm to apply the ADS-MVD mode is depicted in Fig. 5.

Table3. BD-rate Reduction for each sequence in RA

Seq	Y	U	V	Enc	Dec
City	-0.15%	1.27%	0.15%	104%	98%
Crew	-0.07%	-0.24%	0.64%	103%	97%
Vidyo1	-0.19%	0.13%	-0.41%	106%	98%
Vidyo3	-0.40%	0.08%	0.12%	105%	96%
BastDv	-0.15%	-0.20%	0.03%	101%	98%
Cactus	-0.25%	-0.41%	0.03%	102%	97%
MarkPl	-0.45%	-0.11%	-0.23%	103%	97%
RitDac	-0.06%	-0.18%	-0.21%	100%	97%
Tango2	-0.07%	0.06%	0.24%	104%	99%
Campf	-0.01%	-0.03%	0.05%	100%	98%
ParkRu	-0.36%	-0.25%	-0.25%	100%	99%
DayLR2	-0.22%	-0.20%	0.05%	103%	100%
720p	-0.20%	0.31%	0.12%	105%	97%
1080p	-0.23%	-0.22%	-0.09%	101%	97%
4K	-0.16%	-0.11%	0.02%	102%	99%
All	-0.20%	-0.01%	0.02%	103%	98%



(a) HPM-5.0 as the anchor



(b) HPM-5.0 with proposed method

Fig. 6. Best modes for each CU in HPM-5.0 as the anchor (a) and HPM-5.0 with the proposed algorithm (b). Blue borders represent CUs with UMVE mode, green borders represent CUs with affine mode (including affine AMVP mode and normal affine direct/skip mode) and red borders represent CUs with the proposed mode.

4. EXPERIMENTAL RESULTS

The proposed method is implemented on the AVS-3 reference software HPM-5.0 [15] to evaluate its coding performance. The test configuration is random access (RA) and the simulations are performed following the common test conditions (CTC) [16] defined in the development of AVS-3 phase2. HPM-5.0 is regarded as the anchor. Table 3 illustrates the overall results in terms of BD-rate [17] and the encoding/decoding time changing.

It can be observed, compared with HPM5.0, the proposed ADS-MVD can achieve 0.20% BD-rate reduction in RA configuration, with 3% encoding time increasing and no decoding time changing. In addition, it should be noted that the proposed method can improve the coding performance on all sequences with different resolutions, bit-depths and contents.

For the sequences with rich affine motions, coding gains are much higher. Comparing with sequences with intense affine motions, sequences with slight or moderate affine motions can achieve more coding gains. For example, the coding gain on *ParkRunning3* sequence is 0.36% but that on *Campfire* sequence is 0.01%. That is because the limited predefined CPMVD sets are more suitable to represent affine motion in a limited range.

The encoding time is increased mainly because of the extra RDO checking. The decoding time is reduced slightly on account that fewer residuals need to be decoded with a higher prediction quality.

To get a better understanding of the proposed method, the best mode of each CU in HPM-5.0 as the anchor and HPM-5.0 with the proposed algorithm are visualized in Fig. 6. Fig. 6(a) shows CUs with the UMVE mode (blue borders) and CUs with the affine mode (green borders) in HPM-5.0 as the anchor. Fig. 6(b) adds CUs with ADS-MVD (red borders). Comparing Fig. 6(a) and Fig. 6(b), it can be seen that the regions where CUs with UMVE mode and affine mode are selected alternately with almost equal chances in HPM-5.0, such as in the center area of the tablet in Fig. 6, is in favor of the proposed mode. In those areas, the trade-off between reconstruction quality and overhead bits provided by the proposed mode is better than that of any other existing mode in HPM-5.0.

5. CONCLUSION

In this paper, ADS-MVD is proposed to extend the concept of UMVE to AMC. The proposed method refines CPMVs of an affine direct/skip candidate with one of the predefined

CPMVDs. Five predefined MVD magnitudes and four predefined MVD directions are available for each control point. A better trade-off between the precision of AMM and the consumed bits on CPMVDs can be achieved with the proposed method. Simulation results verify the effectiveness of the proposed method and show that 0.20% coding gain on average is obtained. Therefore, the proposed method has been adopted as a coding tool in AVS-3 phase2.

6. ACKNOWLEDGE

This work was done while the author was a research intern in ByteDance Inc.

7. REFERENCES

[1] "Gb/t 20090.2 information technology advanced audio video coding standard part 2: Video," 2006.
[2] "Gb/t 33475.2 information technology efficient multimedia coding standard part 2: Video," 2016.
[3] K. Fan, X. Xie, Z. Wang, G. Xu, R. Wang, X. Lu, H. Chen, Y. Zhao, and H. Yang, "HPM: new reference software platform for avs3," AVS workgroup Doc. M4510, Feb. 2018.
[4] M. Wang, J. Li, L. Zhang, K. Zhang, H. Liu, S. Wang, S. Kwong and S. Ma, "Extended quad-tree partitioning for future video coding," *Data Compression Conference*, pp. 300–309, Mar. 2019.
[5] G. Xu, K. Fan, and R. Wang, "Intra dual prediction," AVS workgroup Doc. M4609, Dec. 2018.
[6] J. Li, M. Wang, L. Zhang, K. Zhang, H. Liu, J. Xu, T. Fu, Y. Wang, P. Zhao, D. Hong, and S. Wang, "Chroma
Coding with two-step cross-component prediction," AVS workgroup Doc. M4632, Dec. 2018.
[7] X. Lu and H. Yang, "Affine motion compensation," AVS workgroup Doc. M4451, Aug. 2018.
[8] X. Ouyang, F. Wang, J. Chen, Y. Piao, and K. Choi, "CE7: Ultimate motion vector expression," AVS workgroup Doc. M4512, Dec. 2018.
[9] J. Li, L. Zhang, K. Zhang, H. Liu, Y. Wang, P. Zhao, and D. Hong, "History-based motion vector prediction for AVS," AVS workgroup Doc. M4488, Aug. 2018.
[10] F. Wang, X. Ouyang, J. Chen, Y. Piao, and K. Choi, "CE6: Adaptive motion vector resolution," AVS workgroup Doc. M4525, Dec. 2018.
[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] D. Zhao, N. Zhang, X. Fan 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] D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, L. Li, H. Li 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] "AVS-3 software platform-5.0," ftp://47.93.196.121.
[16] K. Fan, "AVS3-p2 common test conditions v6.0," AVS workgroup Doc. N2654, Mar. 2019.
[17] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG.16 Q.6 VCEGM33, 2001.