Recent Development of AVS Video Coding Standard: AVS3

Jiaqi Zhang*†, Chuanmin Jia‡, Meng Lei‡, Shanshe Wang‡, Siwei Ma‡, and Wen Gao‡
*Institute of Computing Technology Chinese Academy of Sciences, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China
Email: zhangjiaqi17@mails.ucas.ac.cn
‡Institute of Digital Media, Peking University, Beijing, China
Email: {cmjia, leimeng, sswang, swma, wgao}@pku.edu.cn

***Abstract*—The paper presents a systematic and comprehensive overview of the third generation of Audio Video Standard (AVS) in China. The AVS standards continuously attract extensive attention both domestically and worldwide along with the industrialization of AVS2 standard and the broadcasting of China Central Television (CCTV) 4K channel. Currently, AVS3 has adopted many novel coding techniques including block partitioning structure, intra/inter and transform coding tools. The compression performance has been significantly improved that the latest version of AVS3 obtains about 24.3% and 26.88% bit-rate reduction against AVS2 and HEVC respectively under 4K resolution sequences.**

***Index Terms*—AVS3, video coding standard, intra coding, inter coding**

I. Introduction

To adapt to the growing demand for ubiquitous Ultra High Definition (UHD) contents, e.g. 8K, the AVS working group officially initiated the next generation video coding standard, AVS3, in March 2018, targeting at achieving significant bit-rate saving with negligible quality degradation over AVS2. And the first phase of AVS3 has been finalized in March, 2019. Successive exploration is also planned to enhance the standard to the subsequent phases to further improve the coding performance.

The AVS3 standard is built on the top of its predecessor, AVS2 [1], which was initially developed between 2012 and 2015. However, with the increasing demand and diversity of video services, as well as the growing popularity of UHD and virtual reality (VR) multimedia contents, AVS2 shows less capability when handling such videos. Hence, the design philosophy of AVS3 has been focused on two major categories: higher video resolution contents coding and more efficient parallel processing architectures.

Generally, AVS3 adopts block-based hybrid video coding framework, including prediction, transform/quantization, entropy coding and in-loop filtering. The AVS working group has been working on not only the development of the text specification document, but also the reference software of AVS3 for the standardization process. In this paper, we provide a comprehensive overview of the novel coding tools and the algorithmic design details for better understanding of the AVS3 standard.

This work was supported by the MoE-China Mobile Research Fund Project(MCM20180702), and High-performance Computing Platform of Peking University, which are gratefully acknowledged.

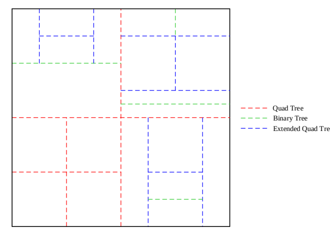

Fig. 1. New Coding Unit Shape in AVS3

II. Flexible Coding Unit Partition in AVS3

On the top of AVS2, AVS3 adopts a more flexible block partition structure to support various resolution video contents, especially for the UHD videos. In AVS2, only quadtree partition (QT) is utilized [2], which is less efficient for 8K videos. AVS3 adopts a more flexible partition mechanism, including binary tree (BT) partition [3] and extended quadtree (EQT) partition [4], which is shown in Fig. 1. While in Fig. 2, the partitioning detail for the largest coding unit (LCU) is depicted. For each LCU, one bit is first used to indicate whether the LCU is divided into QT or not. If not, one more bit is then used to represent whether the LCU is divided into EQT or BT. Note that once the coding unit is divided into BT or EQT, QT cannot be used in the subsequent partition process.

III. Key Coding Tools for Intra Coding in AVS3

Regarding the intra coding in AVS3, some new coding tools including intra derived tree (Intra-DT), intra prediction filter (IPF), two step cross-component prediction mode (TSCPM) are specially designed.

A. Intra-DT

Intra Derived Tree (Intra-DT) [5] defines intra coding unit (CU) partitioning mechanism vertically or horizontally. The

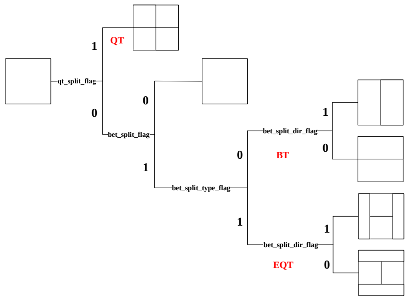

Fig. 2. The Coding Structure for A LCU in AVS3

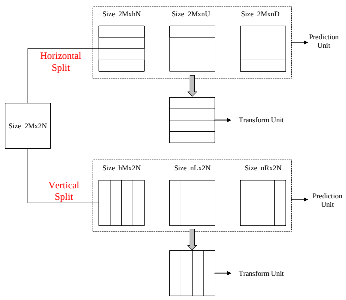

Fig. 3. The Split Manner of Intra-DT

current CU is divided into two or four prediction units (PU) when the current CU apply to intra-DT. As shown in Fig. 3, intra-DT has 6 split types. Transform unit (TU) would also be forced split according to the split direction when intra-DT is used by current CU. The intra-DT flag is signalled at CU level when the intra-DT is used by current CU. The split direction and split type are subsequently transmitted when the intra-DT flag is true.

B. IPF

Intra Prediction Filter (IPF) [6] [7] is a coding tool to improve the accuracy of the intra predicted pixels. The predicted value near the top and left boundary of current CU is filtered to enhance the prediction accuracy. The detail is shown in Fig. 4. The syntax element to switch IPF is designed at CU level by one bit. In particular, the bit only needs to be transmitted when intra-DT is not used in current CU. The filter coefficients are determined according to the following conditions: the shape of the prediction block, the distance between the prediction and the reference sample. Specifically, the process of IPF is divided into three cases according the prediction mode as follow:

- For DC, Plane and Bilinear mode, a column of reconstructed pixels on the left and a row of reconstructed pixels on the top is used as reference pixels, and a 3-tap filter is designed for the filtering.
- For the vertical prediction modes in HPM (modes 3 to 18), the reconstructed pixels in the left column are used as reference pixels, and a 2-tap filter is used.
- For the horizontal prediction modes in HPM (mode 19 to 32), the reconstructed pixels in the upper row are used as reference pixels, and a 2-tap filter is used.

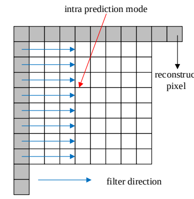

Fig. 4. Intra Prediction Filter

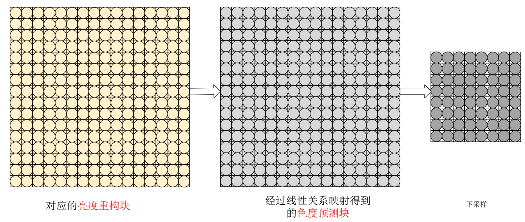

Fig. 5. The Frediction Process of TSCPM

C. TSCPM

Two-Step Cross-component Prediction Mode (TSCPM) [8] [9] is designed for intra chroma coding, which assumes the linear correlation between the prediction results of luma and chroma components. As shown in Fig. 5, the linear regression function is first applied to the originally reconstructed luma block to get an internal prediction block. Second, the internal prediction block is down-sampled to generate the final chroma prediction block. In addition, the parameter α and β of the linear model are derived based on the four points according to the availability of the neighboring samples.

IV. KEY CODING TOOLS FOR INTER CODING IN AVS3

A. HMVP

A History-based Motion Vector Prediction (HMVP) [10] method is employed by AVS3 for inter coding. The HMVP table is established and simultaneously updated according to coding order with historical motion information, which can be directly accessed from previous coded blocks.

As shown in Fig. 6, the HMVP table is first constructed and then updated after processing each coding block. The updating process keeps the same between encoder and decoder. Regarding the update strategy, as illustrated in Fig. 7, the associated motion information is appended to the tail of the HMVP table without redundancy. If the same HMVP candidate is detected in the HMVP table then it will be removed from the current position and appended to the tail of the HMVP table. The HMVP candidates could be applied to

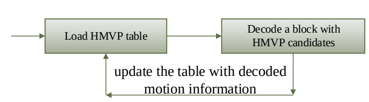

Fig. 6. Decode Flow Chart with HMVP

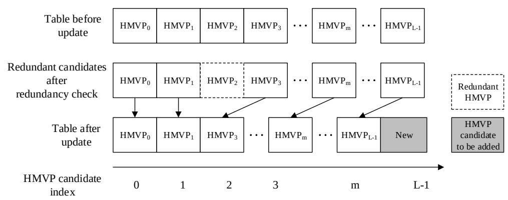

Fig. 7. Update HMVP Table

TABLE I
AMVR INDEX AND CORRESPONDING CODEWORD

AMVR Index	0	1	2	3	4
Resolution (R) in pixel	1/4	1/2	1	2	4
Codeword	0	10	110	1110	1111

the skip/direct mode as complementary candidates in addition to those traditional candidates derived from spatial or temporal neighboring blocks. The maximum size of HMVP table is 8 which should be specified in sequence header.

B. AMVR

Adaptive Motion Vector Resolution (AMVR) [11] is adopted to reduce the coding bits for the motion vector difference (MVD) coding. The MVD is usually coded with 1/4-pel precision. While AMVR allows various resolutions for MVD coding. For instance, 1/4–pel, 1/2–pel, 1–pel, 2–pel and 4–pel resolution are supported. Notably, the AMVR flag is signalled at CU level and the AMVR index is also sent indicating which resolution should be used. Since AVS3 supports multiple motion vector resolutions. the selection of the motion vector resolution is signalled at the CU level when CU is inter mode. Based on the resolution, MVD (MVDx, MVDy) is first aligned to such resolution and then signalled according to the following:

$$MVD_x = \frac{MV_x - MVP_x}{R*4}, MVD_y = \frac{MV_y - MVP_y}{R*4} \quad (1)$$

It should be noted that MVP is also need to be aligned to the resolution of MVD. Table I shows the AMVR index and the corresponding codewords. Four context models are used, one for each position of the bin.

C. AFFINE

In AVS3, Affine prediction mode considers more complex motion instead of translational motion. There are two types of affine motion prediction, one is affine direct mode [12] while the other is affine inter mode [13].

1) Affine direct mode: Affine direct mode can be applied for CUs with both width and height larger than or equal to 16. In affine direct mode, an affine candidate list including at most 5 affine models is derived from the motion information of neighboring blocks, and an index is signalled in the bitstream to get the specified affine model from the affine candidate list at the decoder side. An affine model candidate is a 4-parameter affine model or 6-parameter affine model, represented by 2 or 3 motion vectors (called control point motion vectors, CPMV) of the current block. The following three types of CPVM candidate are used to form the affine merge candidate list:

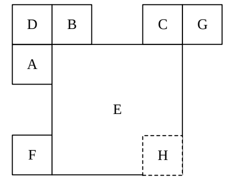

Fig. 8. Neighboring Block

- ***Inherited affine candidates***
  Inherited affine direct candidate means that the candidate is derived from the affine motion model of its available neighboring affine coded block. In AVS3, up to two different neighboring coding blocks are used to derive the candidates. As shown in Fig. 8, the scan order is F, G, C, A, D. The CPMV candidate in the affine direct list of the current CU are derived based on the affine motion type and CPMV of an affine CU in the first two available block.
- ***Constructed affine candidates***
  Constructed affine candidate means the candidates are constructed by combining the neighbor motion information of each control point. First, the motion information for the control points of current CU is derived from the specified spatial neighbors and temporal neighbor shown in Fig. 8. MVXk (k=0, 1, 2, 3) represents the k-th control point. F, G, C, A, B and D are spatial positions for predicting MVXk (k=0, 1, 2); H is temporal position for predicting MVX3. Second, the combinations of control points are used to construct an affine merge candidate. If the reference indices of control points are different or control points can not be obtained, the constructed motion model is discarded.
- ***Zero MVs***
  If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are inserted into the candidate list, until the list is full.

2) Affine inter mode: An affine flag is signalled in the bitstream at CU level to indicate whether affine inter mode is used. When the CU is affine inter mode, it has two CPMV, in other word, it is 4-parameter affine motion model. In affine inter mode, one CPMV is obtained from A, B, D in Fig. 8, which is similar to derive MVP. The other CPMV is obtained from G, C.

D. UMVE

Ultimate motion vector express (UMVE) [14] technology is an effective way to express motion vector for skip and direct modes in AVS3. According to the base motion vectors derived

TABLE II
THE CODEWORD OF MV OFFSET DIRECTION

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

TABLE III
THE CODEWORD OF MV OFFSET DIRECTION

Distance Index	0	1	2	3	4
Pel distance	1/4-pel	1/2-pel	1-pel	2-pel	4-pel

from the information of spatial or temporal neighboring candidates, this technique offsets these motion vectors to obtain the motion vectors that are more suitable for the current coding blocks. Two base motion vectors can be derived at most. When the neighboring CU is inter mode, the motion information of these CU can be used to derive base motion vector. The scan order of the spatial neighboring block is F, G, C, A, D, Fig. 8 show the position of each candidate. Temporal motion information and zero MV will be used when the spatial motion information is not enough.

The specific offset information of the base motion vector is determined by the encoder and transmitted to the decoder in the bitstream. According to the information in the bitstream, the decoder derives the motion vector for decoding. The UMVE flag is signalled at CU level when the mode of current CU is skip or direct mode. The other information about the motion vector that UMVE technology needs to transmit includes: scan order of neighboring block, MV offset direction and MV offset distance. MV adjustment direction is limited in horizontal and vertical, Table II shows its codeword. MV offset distance index, as shown in Table III, indicates MV adjustment distance. A total of five adjust distance, 1/4–pel, 1/2–pel, 1–pel, 2–pel and 4–pel, are used in UMVE. Fig. 9 show the position which adjust mv might be locate.

E. EMVR

Based on the above HMVP and AMVR, an enhanced motion vector resolution (EMVR) mode [15] is carefully designed to avoid the shortcoming of one motion vector prediction (MVP) compared to multiple MVP. EMVR is proposed to bind the AMVR and HMVP, which corresponds the existing AMVR index of current CU with the motion information in HMVP table, and then conducted inter-frame prediction. The corresponding relationship between AMVR index and HMVP is shown in the Fig. 10.

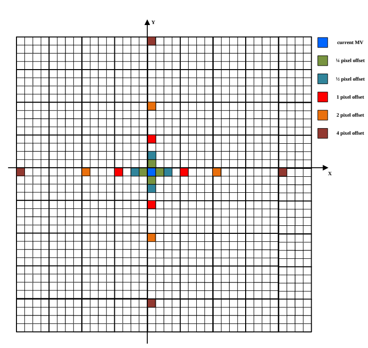

Fig. 9. Position of Adjust MV

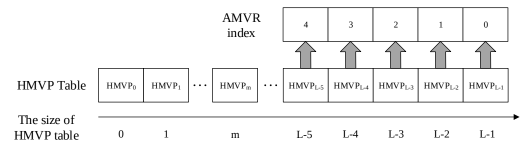

Fig. 10. Relationship of AMVR Index and HMVP Candidate

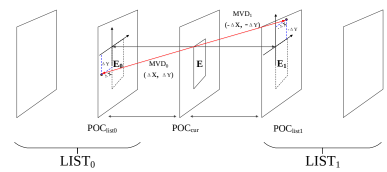

Fig. 11. Symmetry Motion Vector Difference

When candidate motion information of HMVP table corresponding to AMVR Index is selected, if the motion information only has MV0 motion vector of List0, the MV0 scale of this motion vector is required to be on List1, and then other steps of inter-frame prediction are carried out. If the motion information is only the motion vector MV1 of List1, the scale of this motion vector MV1 is required to be on List0, and then other steps of inter-frame prediction are carried out. At the decoder side, we firstly determined whether use EMVR or not, original AVS MVP will be constructed if EMVR is not used. Otherwise, the MVP should be obtained from HMVP table according to the AMVR index. When the motion information in HMVP table is not insufficient, the last entry in HMVP table will be used. When HMVP table is empty, the motion vector and reference index is set to be 0.

F. SMVD

Symmetry motion vector difference (SMVD) [16] is used for motion vector difference in inter Bi-directional prediction mode. One bit is used to indicate whether SMVD is adopted when the prediction mode of current CU is Bi-directional prediction mode. When SMVD is used, the motion information in List1 is symmetrical with respect to List0. As illustrate in Fig. 11, MVD1 of List1 is set to -MVD0 accordingly. Reference index values of both List0 and List1 are set to 0 and don‘t need to be transmitted. Relationship among current frame, forward and backward reference frame is described as follow:

$$POC_{cur} - POC_{list0} = POC_{list1} - POC_{cur} \tag{2}$$

In the above formula, POC is acronym of picture order count.

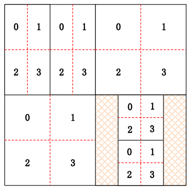

Fig. 12. CU Split by PBT

TABLE IV
PRE-DESIGNED TRANSFORM CORE

Position Index	Horizontal Transform	Vertical Transform
0	DCT-VIII	DCT-VIII
1	DST-VII	DCT-VIII
2	DCT-VIII	DST-VII
3	DST-VII	DST-VII

V. KEY CODING TOOLS FOR TRANSFORM IN AVS3

A. PBT

Position based transform (PBT) [17] introduces a new transform method which uses pre-designed transform sets for four sub-blocks according to their positions. A new CU-level syntax element, PBT flag, is introduced. When PBT flag is 0, the traditional DCT-II transform is used for current CU. When PBT flag is 1, PBT splits the CU into four sub-blocks, and use pre-designed transform set according to their position. As shown in Fig. 12, if a CU is applied with PBT, it is split by red dotted lines into four sub-blocks, and the orange grid region represent the block can‘t use PBT. In each sub-block, its position number is connected with a pre-designed transform set shown in Table IV. Note that PBT can only be applied to inter mode and luma CBs with size ranging from 8x8 to 32x32, and the ratio of width and height should less than 4. Besides, PBT flag is coded together with coded block flag (cbf). When PBT flag is 0, a CU is consisted of a luma TU and two chroma TUs. Otherwise, a CU is consisted of four luma TUs and two chroma TUs.

VI. EXPERIMEANTAL RESULTS

In this section, we provide the coding performance of AVS3 coding standard (phase 1) under common test condition [18]. AVS2 and HEVC are utilized as anchor. The reference software of AVS3, AVS2 and HEVC is HPM3.2, RD19.50 and HM16.20 respectively.

A. R-D performance

As shown in Table V, the coding performance under random access (RA) configuration is reported.

The experimental results show that the AVS3 achieves on average 23.77% and 21.75% bd-rate reduction against AVS2 and HEVC respectively under RA configuration.

TABLE V
BD-RATE PERFORMANCE OF AVS3 (ANCHOR: AVS2 AND HEVC)

Class	VS AVS2			VS HEVC		
	Y(%)	U(%)	V(%)	Y(%)	U(%)	V(%)
UHD4K	-24.30%	-27.30%	-29.53%	-26.88%	-18.87%	-21.71%
1080p	-23.64%	-27.73%	-28.67%	-21.58%	-18.53%	-18.86%
720p	-23.36%	-27.24%	-26.76%	-16.80%	-17.41%	-16.32%
Average	-23.77%	-27.42%	-28.32%	-21.75%	-18.27%	-18.96%

VII. CONCLUSION

The emerging AVS3 is the state-of-the-art video coding standard developed by AVS working group. Targeting at efficiently coding for UHD and VR contents, AVS3 adopted a number of novel coding tools to promote coding efficiency. AVS3 phase one achieves 23.77% bd-rate reduction over AVS2 which sets up a milestone for AVS3 standard. Moreover, subsequent efforts and explorations are scheduled to enhance the standard and obtain higher performance.

REFERENCES

[1] Wen Gao and Siwei Ma. An overview of avs2 standard. In *Advanced Video Coding Systems*, pages 35–49. Springer, 2014.
[2] Yu Qin, Ma Siwei, He Zhichu, Li Weiran, Gao Min, Zheng Xiaozhen, and Kim Il-Koo. Suggested video platform for avs2. *AVS-Doc, M2972*, 2012.09.
[3] Wang LiQiang, Niu BenBen, Wei ZiWei, Lei Meng, and Zhang JiaQi. Flexible block partitioning structure based on tavs3 platform.
[4] Wang Meng, Li Junru, Zhang Li, Zhang Kai, Liu Hongbin, Wang Yue, Zhao Pengwei, Hong Dingkun, and Wang Shiqi. Extended quad-tree partitions. *AVS-Doc, M4507*, 2018.10.
[5] Wang Liqiang, Niu Benben, Wei Ziwei, Xiao Haodong, and He Yun. Ce1 : derived mode. *AVS-Doc, M4540*, 2018.12.
[6] Xu Guisen, Fan Kui, and Wang Ronggang. Intra prediction filter. *AVS-Doc, M4609*, 2018.12.
[7] Xu Guisen, Fan Kui, and Wang Ronggang. Supplementary test for m4609. *AVS-Doc, M4618*, 2018.12.
[8] Li Junru, Wang Meng, Zhang Li, Zhang Kai, Liu Hongbin, Xu Jizheng, Fu Tianliang, Piao Yinji, and Choi Kiho. Chroma coding with two-step cross-component prediction. *AVS-Doc, M4632*, 2018.12.
[9] Li Junru, Wang Meng, Zhang Li, Zhang Kai, Liu Hongbin, Xu Jizheng, Fu Tianliang, Wang Yue, and Wang Shiqi. Modified comparison logic for tscpm. *AVS-Doc, M4726*, 2019.03.
[10] Li Junru, Zhang Li, Zhang Kai, Liu Hongbin, and Wang Yue. History-based motion vector prediction for avs3. *AVS-Doc, M4488*, 2018.08.
[11] Wang Fan, Ouyang Xiao, Chen Jie, Piao Yinji, and Choi Kiho. Ce6 : The report of adaptive motion vector resolution. *AVS-Doc, M4466*, 2018.08.
[12] Chen Huanbang, Zhao Yin, and Yang Haitao. Affine skip mode. *AVS-Doc, M4643*, 2019.01.
[13] Lu Xiaomu and Yang Haitao. Ce8 : The integration progress report of affine motion model in tavs3. *AVS-Doc, M4451*, 2018.08.
[14] Ouyang Xiao, Wang Fan, Chen Jie, Piao Yinji, and Choi Kiho. Ce7 : The report of ultimate motion vector expression. *AVS-Doc, M4512*, 2018.12.
[15] Ouyang Xiao, Wang Fan, Lv Zhuoyi, Piao Yinji, and Choi Kiho. The improvement of inter prediction mode. *AVS-Doc, M4661*, 2019.01.
[16] Chen Huanbang and Yang Haitao. The performance of smvd on hpm2.2. *AVS-Doc, M4680*, 2019.01.
[17] Wang Liqiang, Niu Benben, Xiao Haodong, Wei Ziwei, and He Yun. Ce4 : Position based inter prediction residual transform method. *AVS-Doc, M4541*, 2018.12.
[18] Fan Kui. Avs3-p2 common test condition. *AVS-Doc, N2654*, 2019.03.