DOI: 10.3969/j. issn. 1673-5188. 2016. 01. 001
http://www.cnki.net/kcms/detail/34.1294.TN.20160202.1459.002.html, published online February 2, 2016

Overview of the Second Generation AVS Video Coding Standard (AVS2)

Shanshe Wang[1], Falei Luo[2], and Siwei Ma[1]
(1. Peking University, Beijing 100871, China;
2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

Abstract

AVS2 is a new generation video coding standard developed by the AVS working group. Compared with the first generation AVS video coding standard, known as AVS1, AVS2 significantly improves coding performance by using many new coding technologies, e.g., adaptive block partition and two level transform coding. Moreover, for scene video, e.g. surveillance video and conference video, AVS2 provided a background picture modeling scheme to achieve more accurate prediction, which can also make object detection and tracking in surveillance video coding more flexible. Experimental results show that AVS2 is competitive with High Efficiency Video Coding (HEVC) in terms of performance. Especially for scene video, AVS2 can achieve 39% bit rate saving over HEVC.

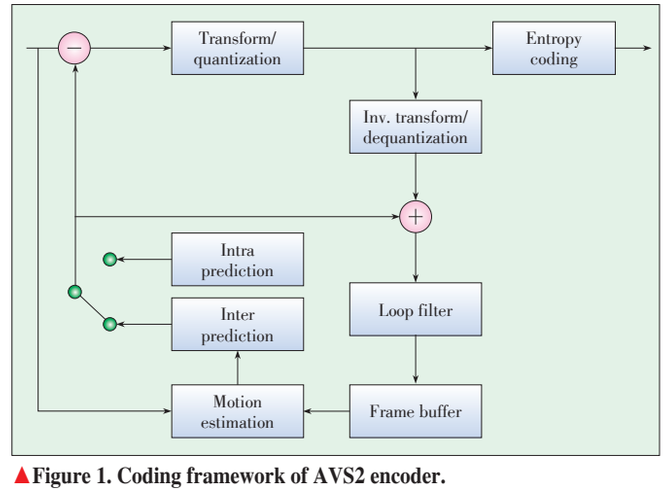
Keywords

Video Coding; AVS2; AVS1

1 Introduction

AVS1 video coding standard, developed by AVS working group, has achieved great success in China and has become an international video coding standard. However, with increased demand for high-resolution videos, it is necessary to develop a new video coding standard that provides much higher coding performance. Based on the success of AVS1 and recent video coding research and standardization, the AVS working group has started the second generation video coding standardization project from 2012, called AVS2. AVS2 is designed to improve coding efficiency for higher resolution videos, and to provide efficient compression solutions for various kinds of video applications, e. g., surveillance video and conference video.

As with previous coding standards, AVS2 uses the traditional hybrid prediction/transform coding framework (**Fig. 1**). However, AVS2 has more flexible coding tools to satisfy the new requirements identified from emerging applications. First, more flexible prediction block partitions are used to further improve the intra- and inter-prediction accuracy, e.g., square and non-square partitions, which are more adaptive to image content, especially at edge areas. Second, a flexible reference management scheme is proposed to improve inter prediction accuracy. Related to the prediction structure, the transform block size is more flexible and can be up to 64 x 64 pixels. After transformation, context adaptive arithmetic coding is used for the entropy coding of the transformed coefficients. And a two-level coefficient scan and coding method is adopted to encode the coefficients of large blocks more efficiently. Moreover, for low delay communication applications, e.g., video surveillance, video conferencing, where the background usually does not change often, a background picture model based coding method is developed in AVS2. A background picture constructed from original pictures is used as a reference picture to improve prediction efficiency. Experimental results show that this background



▲**Figure 1. Coding framework of AVS2 encoder.**

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P1

-picture-based prediction coding can improve the coding efficiency significantly. Furthermore, the background picture can also be used for object detection and tracking for intelligent surveillance video coding.

This paper gives an overview of AVS2 video coding standard and a performance comparison with others. The paper is organized as follows. Section 2 introduces the flexible coding structure in AVS2. Section 3 gives an overview of key tools adopted in AVS2. The specially developed scene video coding is shown in Section 4. Section 5 provides the performance comparison between AVS2 and other state-of-the-art standards. Finally, Sections 6 concludes the paper.

2 Flexible Coding Structure in AVS2

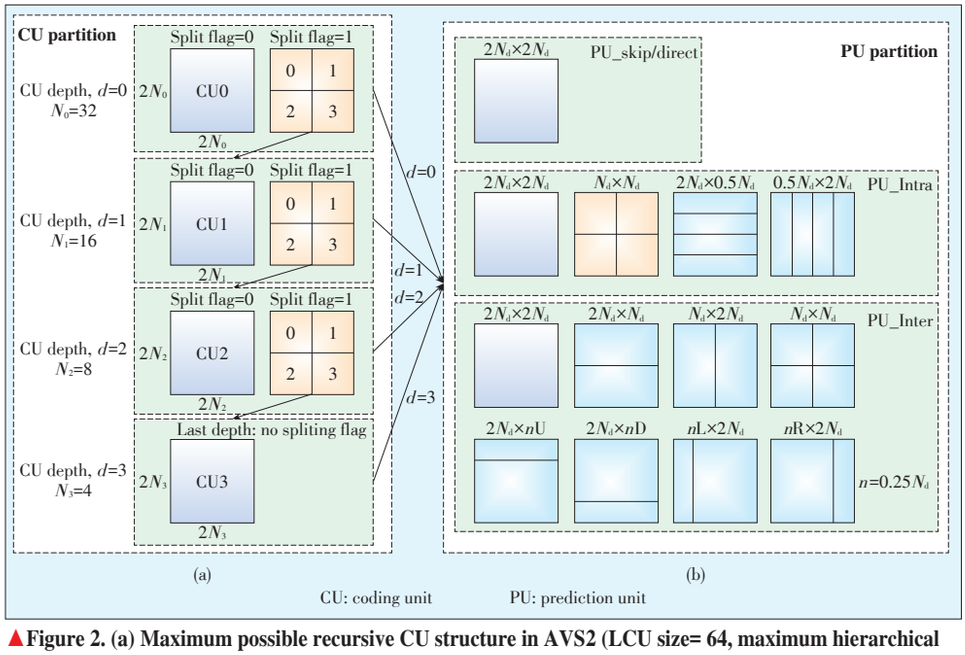
In AVS2, a flexible coding unit (CU), prediction unit (PU) and transform unit (TU) based coding/prediction/transform structure is used to represent and organize the encoded data [1], [2]. First, pictures are split into largest coding units (LCUs), which consist of $2N$ x $2N$ samples of luminance component and associated chrominance samples with N = 8, 16 or 32. One LCU can be a single CU or can be split into four smaller CUs with a quad-tree partition structure. A CU can be recursively split until it reaches the smallest CU size (**Fig. 2a**). Once the splitting of the CU hierarchical tree is finished, the leaf node CUs can be further split into PUs. A PU is the basic unit for intra- and inter-prediction and allows different shapes to encode irregular image patterns (**Fig. 2b**). The size of a PU is limited to that of a CU with various square or rectangular shapes. Specifically, both intra- and inter-prediction partitions can be symmetric or asymmetric. Intra-prediction partitions vary in the set {$2N$x$2N$, NxN, $2N$x$0.5N$, $0.5N$x$2N$}, and inter-prediction partitions vary in the set {$2N$x$2N$, $2N$xN, Nx$2N$, $2N$xnU, $2N$xnD, nLx$2N$, nRx$2N$}, where U, D, L and R are the abbreviations of Up, Down, Left and Right respectively. n is equal to $0.25N$. Besides CU and PU, TU is also defined to represent the basic unit for transform coding and quantization. The size of a TU cannot exceed that of a CU, but it is independent of the PU size.

3 Main Coding Tools in AVS2

AVS2 uses more efficient coding tools to make full use of the textual information and spatial/temporal redundancies. These tools can be classified into four categories: 1) prediction coding, including intra prediction and inter prediction; 2) transform; 3) entropy coding; and 4) in-loop filtering.

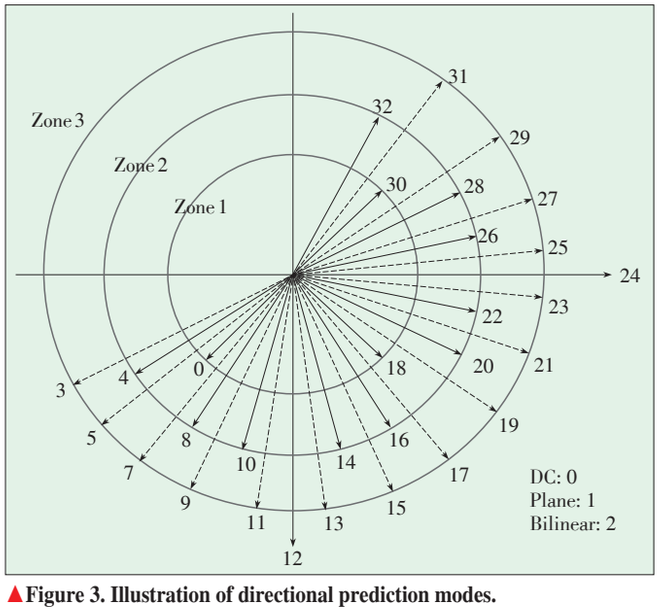
3.1 Intra Prediction

AVS2 still uses a block-partition-based directional prediction to reduce the spatial redundancy in the picture [3]. Compared with AVS1, more intra coding modes are designed to improve the prediction accuracy. Besides the square PU partitions, non-square partitions, called short distance intra prediction (SDIP), are used by AVS2 for more efficient intra luminance prediction [4], where the nearest reconstructed boundary pixels are used as the reference sample in intra prediction (Fig. 2). For SDIP, a $2N$x$2N$ CU is horizontally or vertically partitioned into four PUs. SDIP is more adaptive to the image content, especially in areas with complex textures. To reduce complexity, SDIP is disabled for a 64 x 64 CU. For each prediction block in the partition modes, 33 prediction modes are supported for luminance, including 30 angular modes [3], plane mode, bilinear mode and DC mode. As in **Fig. 3**, the prediction directions associated with the 30 angular modes are distributed within the range of [-157.5°, 60°]. Each sample in a PU is predicted by projecting its location to the reference pixels in the selected prediction direction. To improve intra-prediction accuracy, the sub-pixel precision reference samples are interpolated if the projected reference samples locate on a non-integer position. The non-integer position is bounded to 1/32 sample precision to avoid floating point operation, and a 4-tap linear interpolation filter is used to obtain the sub-pixel. During the coding of luma prediction mode, two most probable modes (MPMs) are used for



▲**Figure 2. (a) Maximum possible recursive CU structure in AVS2 (LCU size= 64, maximum hierarchical depth = 4), (b) Possible PU splitting for skip, intra and inter modes in AVS2.**

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P2



▲Figure 3. Illustration of directional prediction modes.

prediction. If the current prediction mode equals one of the MPMs, two bins are transmitted into the bitstream; otherwise, six bins are needed.

For the chrominance component, the PU is always square, and 5 prediction modes are supported, including vertical prediction, horizontal prediction, bilinear prediction, DC prediction and the prediction mode derived from the corresponding luminance prediction mode [5].

3.2 Inter Prediction

Compared to the spatial intra prediction, inter prediction focuses on exploiting the temporal correlation between the consecutive pictures to reduce the temporal redundancy. AVS2 still adopts the multi-reference prediction as in AVS1, including both short term and long term reference pictures. However, inter prediction mode has been improved much and a more flexible reference picture management scheme is adopted.

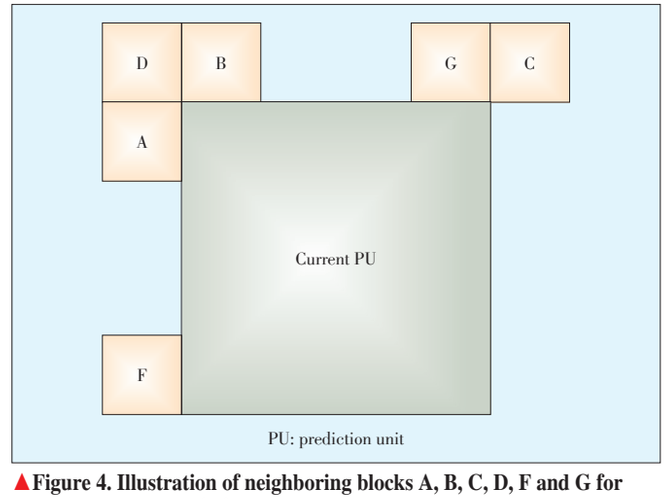
3.2.1 Improved Inter-Prediction Mode

In AVS2, inter prediction mode has been improved much to further improve the inter prediction accuracy. Firstly, a new inter frame type, called F frame, is defined as a special P frame [6] in addition to the traditional P and B frames. Secondly, new inter coding modes are specially designed for F and B frame.

For F frame, besides the conventional single hypothesis prediction mode as in a P frame, the significant improvement is the use of multi-hypothesis techniques, including multi-directional skip/direct mode [7], temporal multi-hypothesis prediction mode [8], and spatial directional multi-hypothesis (DMH) prediction mode [9]. These modes improve the coding performance of AVS2 by a large margin. Detailed descriptions are shown as follows.

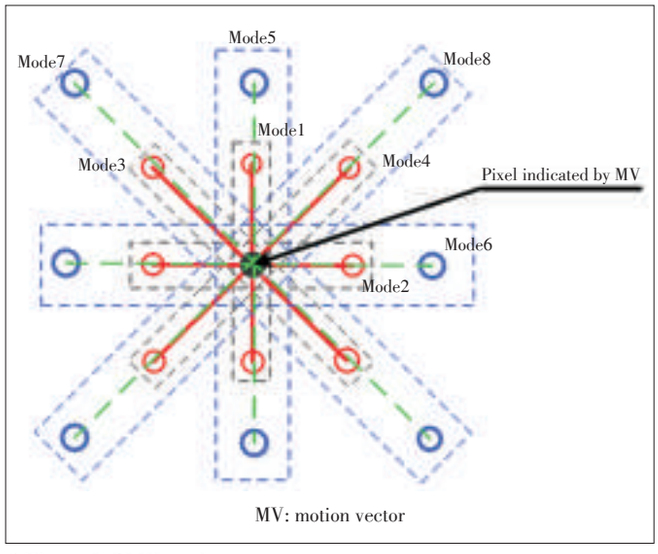
The multi-directional skip/direct mode in F frame is used to merge current block to spatial or temporal neighboring block. The difference between skip mode and direct mode is that skip mode needs to encode residual information while direct mode does not. However, the derivation of motion vector (MV) for the two modes are the same. In AVS2, two derivation methods, one of which is temporal and the other is spatial, are used. For temporal derivation, one MV is achieved from the temporal collocated block in the nearest or backward reference frame. The other MV for weighted skip mode is obtained by scaling the first derived MV in the second reference frame. The second reference is specified by the reference index transmitted in the bitstream, indicating weighted skip mode. For spatial derivation, the needed motion vectors, one or two, are obtained from neighboring prediction blocks. If only one MV is needed, two derivations are provided. One is to search the neighboring blocks (**Fig. 4**) in a pre-defined order: F, G, C, A, B, D. The other is to determine the MV by searching the neighboring blocks in a reverse order. If the derived MVs do not belong to the same block, the two MVs are available. Otherwise, the second MV should be re-derived from the neighboring blocks using dual forward prediction. If two MVs are needed, the derivation scheme is the same as before. The difference is that when the two MVs belong to the same block, the second MV should re-derived by combining one MV single forward prediction searched by the defined order and one MV searched by reversed order.

DMH mode provides a derivation scheme to generate two seed predictors based on the initial predictor obtained from motion estimation to improve the inter prediction accuracy. As in **Fig. 5**, all the optional seed predictors are located on the line crossing the initial predictor. Considering the coding complexity, the number of seed predictors is restricted to 8, mode 1 to mode 8. The derivation of the two seed predictors is shown in **Table 1**. For one seed predictor mode with index as i, MV offset, denoted as $\overrightarrow{dmh_i}$, is firstly obtained according to the table. Then the needed two seed predictors, $\overrightarrow{mv_1}$ and $\overrightarrow{mv_2}$, are calcu-



▲Figure 4. Illustration of neighboring blocks A, B, C, D, F and G for motion vector prediction.

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P3



▲Figure 5. DMH mode.

▼Table 1. The derivation of seed predictors for DMH

Mode index	$\overrightarrow{dmh_i}$
1	(1, 0)
2	(0, 1)
3	(1, -1)
4	(1, 1)
5	(2, 0)
6	(0, 2)
7	(2, -2)
8	(2, 2)

lated based on the original ($\overrightarrow{mv_o}$) as follows.

$$\overrightarrow{mv_1} = \overrightarrow{mv_o} + \overrightarrow{dmh_i} \quad (1)$$

$$\overrightarrow{mv_2} = \overrightarrow{mv_o} - \overrightarrow{dmh_i} \quad (2)$$

For B frame, the coding modes are also expanded to improve prediction accuracy. In addition to the conventional forward, backward, bi - directional and skip/direct prediction modes, symmetric prediction is defined as a special bi - prediction mode, wherein only one forward - motion vector is coded, and the backward motion vector is derived from the forward motion vector.

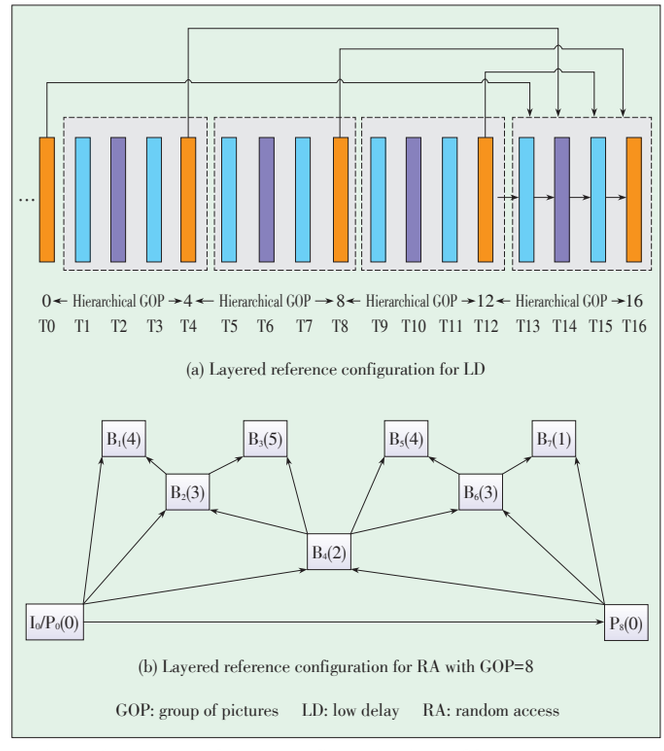
3.2.2 Flexible Reference Picture Management

AVS2 adopts a flexible reference picture management scheme to improve the inter prediction efficiency. In the scheme, a reference configuration set (RCS) is used to manage the reference pictures. RCS consists of reference picture information of current coding picture, including decoding order index (DOI), QP offset, number of reference pictures, delta DOIs between current picture and reference pictures, number of pictures that need to remove from buffer and delta DOIs between pictures to remove and current pictures.

In order to save coding bits, several RCS sets are used and signaled in the sequence header. Only the index of RCS is transmitted in the picture header. Based on RCS, the reference picture set for current coding picture can be arbitrarily configured. **Fig. 6** shows the layered reference configuration on AVS2.

3.3 Motion Vector Prediction and Coding

The motion vector prediction (MVP) plays an important role in inter prediction, which can reduce redundancy between motion vectors of neighbor blocks and save many coding bits for motion vectors. In AVS2, four different prediction methods are adopted (**Table 2**). Each of these has its unique usage. Spatial motion vector prediction is used for spatial derivation of Skip/



▲Figure 6. Layered reference configuration in AVS2.

▼Table 2. MV prediction methods in AVS2

Method	Details
Median	Using the median MV values of the two nearest MVs among scaled MVs of three neighbouring blocks
Spatial	Using the MVs of one spatial neighbouring block
Temporal	Using the scaled MVs of temporal collocated 16x16 block which covers the region of top-left 4x4 block of current prediction unit
Spatial-temporal combined	Using the temporal MVP when the temporal block utilizes inter coding, otherwise using median MV values

MVP: Motion vector prediction

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P4

Direct mode in F frames and B frames. Temporal motion vector prediction is used for temporal derivation of Skip/Direct mode in all inter frames. Spatial-temporal combined motion vector prediction is used for temporal derivation of Skip/Direct mode in B frames. For other cases, median prediction is used. Moreover, in order to improve the MV prediction accuracy, the derivation of MV is achieved by the reference distance based scaling.

In AVS2, the motion vector is in quarter-pixel precision for the luminance component, and the sub-pixel is interpolated with an 8-tap DCT interpolation filter (DCT-IF) [10]. For the chrominance component, the motion vector derived from luminance with 1/8 pixel precision and a 4-tap DCT-IF is used for sub-pixel interpolation [11]. The filter coefficients for sub-pixel interpolation is defined in **Table 3**. After motion vector prediction, the motion vector difference (MVD) is coded in the bitstream. However, redundancy may still exist in MVD, and to further save coding bits of motion vectors, a progressive motion vector resolution (PMVR) adaptation method is used in AVS2 [12]. In PMVR, MVP is first rounded to the nearest half sample position, and then the MVDis rounded to half-pixel precision if it exceeds a threshold. Furthermore, the resolution of MVD is decreased to integer-pel precision if it exceeds another threshold. In AVS2, only one threshold is used, which means that if the distance between the MV and MVP is less than the threshold, quarter-pixel based MVD is coded; otherwise,half-pixel based MVD is coded(actually, the MVD is separated into two parts and coded with different resolution. The part of MVD within the window will be coded at 1/4 pixel resolution, and the other part will be coded at half-pixel resolution).

3.4 Transform

Unlike the transform in AVS1, a flexible TU partition structure is used to further compress the predicted residual in AVS2. For CU with symmetric prediction unit partition, the TU size can be 2*N*x2*N* or *N*x*N* signaled by a transform split flag. For CU with asymmetric prediction unit partition, the TU size can be 2*N*x2*N*, *n*x2*N* or 2*N*x*n*. Thus, the maximum transform size is 64x64, and the minimum is 4x4. For TU size from 4x4 to 32x32, an integer transform (IT) that closely approximates the performance of the discrete cosine transform (DCT) is used. For a square residual block, the forward transform matrices from 4x4 to 32x32. Here,4x4 transform T_4 and 8x8 transform T_8 are:

$$T_4=\begin{bmatrix}32 & 32 & 32 & 32\\ 42 & 17 & -17 & 42\\ 32 & -32 & -32 & 32\\ 17 & -42 & 42 & -17\end{bmatrix} \tag{3}$$

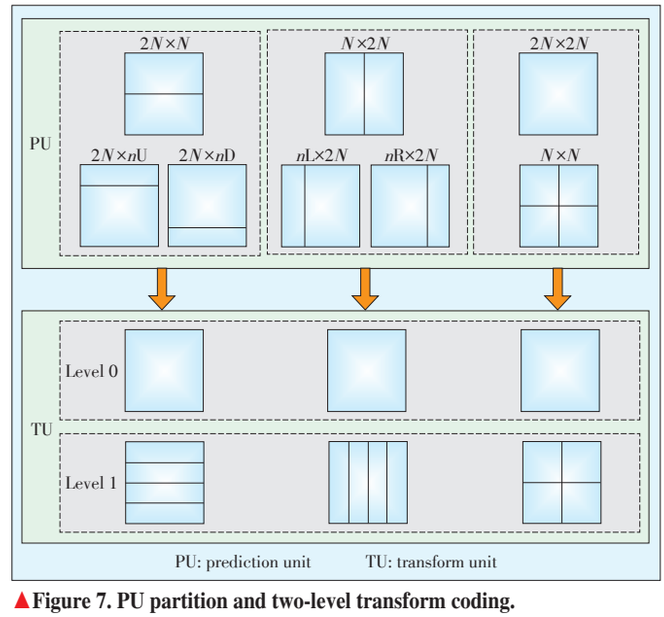
$$T_8=\begin{bmatrix}32 & 32 & 32 & 32 & 32 & 32 & 32 & 32\\ 44 & 38 & 25 & 9 & -9 & -25 & -38 & -44\\ 42 & 17 & -17 & -42 & -42 & -17 & 17 & 42\\ 38 & -9 & -44 & -25 & 25 & 44 & 9 & -38\\ 32 & -32 & -32 & 32 & 32 & -32 & -32 & 32\\ 25 & -44 & 9 & 38 & -38 & -9 & 44 & -25\\ 17 & -42 & 42 & -17 & -17 & 42 & -42 & 17\\ 9 & -25 & 38 & -44 & 44 & -38 & 25 & -9\end{bmatrix} \tag{4}$$

For a 64x64 transform, a logical transform (LOT) [13] is applied to the residual. A 5-3 tap integer wavelet transform is first performed on a 64x64 block discarding the LH, HL and HH-bands, and then a normal 32x32 IT is applied to the LL-band. For all the PU partitions of a CU, 2*N*x2*N* IT is used in the first level, and a non-square transform [14] is used in the second level (**Fig. 7**).

Furthermore, a secondary transform can be used to reduce the correlation for luminance intra-prediction residual block. The secondary transform matrix is related to the block size. If the transform block size is greater than or equal to 8x8, a 4x4 secondary transform with matrix S_4 is applied to the left corner of the transform block as shown in **Fig. 8**. If the transform block size is 4x4, an independent transform matrixD_4 rather

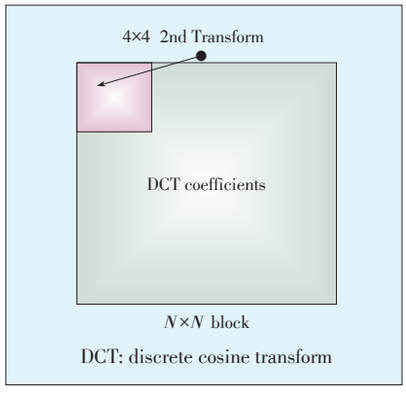
▼**Table 3. DCT-like interpolation filter for sub-pixel interpolation**

Interpolation	Position	Coefficients
Luma	1/4	{ -1, 4, -10, 58, 17, -5, 1, 0}
	2/4	{-1, 4, -11, 40, 40, -11, 4, -1}
	3/4	{0, 1, -5, 17, 58, -10, 4, -1 }
Chroma	1/8	{ -4, 62, 6, 0}
	2/8	{ -6, 56, 15, -1}
	3/8	{ -5, 47, 25, -3}
	4/8	{ -4, 36, 36, -4}
	5/8	{ -3, 25, 47, -5}
	6/8	{ -1, 45, 56, -6}
	7/8	{ 0, 6, 62, -4}



▲**Figure 7. PU partition and two-level transform coding.**

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P5



◀Figure 8. Illustration of secondary transform in AVS2.

than T_4 is used.

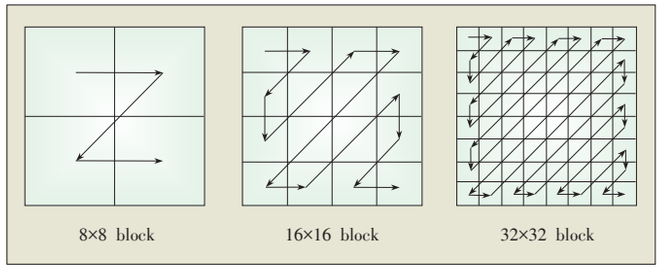
$$S_4=\begin{bmatrix}123 & -35 & -8 & -3\\ -32 & -120 & 30 & 10\\ 14 & 25 & 123 & -22\\ 8 & 13 & 19 & 126\end{bmatrix},\ D_4=\begin{bmatrix}34 & 58 & 72 & 81\\ 77 & 69 & -7 & -75\\ 79 & -33 & -75 & 58\\ 55 & -84 & 73 & -28\end{bmatrix} \quad (5)$$

3.5 Entropy Coding

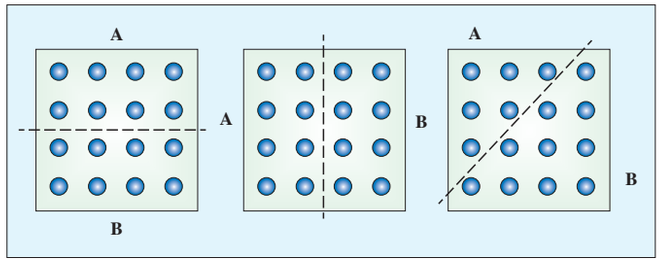
The entropy coding used in AVS2 is inherited form AVS1. The arithmetic coding engine is designed according to a logarithmic model. Thus, the probability estimation is specified to be multiplication-free and only using shifts and addition and no look-up tables are needed.

For the transformed coefficients coding, a two-level coding scheme is applied to the transform coefficient blocks [15]. First, a coefficient block is partitioned into 4x4 coefficient groups (CGs) (**Fig. 9**). Then zig-zag scanning and Context-Based Adaptive Binary Arithmetic Coding (CABAC) is performed at both the CG level and coefficient level. At the CG level for a TU, the CGs are scanned in zig-zag order, and the CG position indicating the position of the last non-zero CG is coded first, followed by a bin string in the reverse zig-zag scan order of significant CG flags indicating whether the CG contains non-zero coefficients. At the coefficient level, for each non-zero CG, the coefficients are further scanned into the form of (run, level) pair in zig-zag order. Level and run indicate the magnitude of a non-zero coefficient and the number of zero coefficients between two non-zero coefficients, respectively. For the last CG, the coefficient position, which denotes the position of the last non-zero coefficient in scan order, is coded first. For a non-last CG, a last run is coded which denotes number of zero coefficients after the last non-zero coefficient in zig-zag scan order. Then the (level, run) pairs in a CG are coded in reverse zig-zag scan order.

For the context modeling, AVS2 uses a mode-dependent context-selection design for intra-prediction blocks [16]. In this context design, 33 intra-prediction modes are classified into three prediction mode sets: vertical, horizontal, and diagonal. Depending on the prediction mode set, each CG is divided to two regions (**Fig. 10**). The intra-prediction modes and CG regions are applied in the context modeling of syntax elements including the last CG position, last coefficient position and run value. In addition, AVS2 takes more consideration on data dependence reduction in context design and explores more possibility for bypass mode as well.



▲Figure 9. Sub-block scan for transform blocks of size 8x8, 16x16 and 32x32 transform blocks; each sub-block represents a 4x4 coefficient group.



▲Figure 10. Sub-block region partitions of 4x4 coefficient group in an intra prediction block.

3.6 In-Loop Filtering

Compared to AVS1, AVS2 has made great improvement over in-loop filtering. Except for de-blocking filter, two more filtering processes are added to AVS2, called sample adaptive offset (SAO) filtering [17] and adaptive loop filter (ALF) [18], to further improve the reconstructed picture quality. Thus in-loop filtering in AVS2 includes the following three sequential procedures: deblocking filtering, SAO and ALF.

The deblocking filter is designed to remove the blocking artifacts caused by block transform and quantization. In AVS2, the basic unit for deblocking filter is an 8x8 block. For each 8x8 block, deblocking filter is used only if the boundary belongs to either of CU boundary, PU boundary or TU boundary. Unlike AVS1, gradient is considered for boundary strength (BS) calculation and then BS is classified into more levels based on the calculated gradient. When the boundary is not the edge of a block which can be CU, PU or TU, BS is set to the lowest value to reduce the complexity.

After the deblocking filter, an SAO filter is applied to reduce the mean sample distortion of a region. The basic unit of SAO is defined as four pixels top-left the LCU region, which is more flexible for parallelization. An offset is added to the reconstructed sample for each SAO filter unit to reduce ringing artifacts and contouring artifacts. There are two kinds of offset

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P6

called Edge Offset (EO) and Band Offset (BO) mode, respectively.

Edge Offset mode first classifies the pixels in the filter unit using four 1-D directional patterns as illustrated in **Fig. 11**. According to these patterns, four EO classes are specified, and only one EO class can be selected for each filter unit. For a given EO class, samples of current filter unit are classified into one of the five categories, which are based on the rules defined in **Table 4**. For pixels in each category except category 0, an offset is derived by calculating the mean of the difference of reconstructed pixel values and original pixel values. The offset and the index of classification pattern are transmitted to a decoder.

Band offset mode classifies the pixels into 32 bands by equally dividing the pixel range. Theoretically, one offset can be derived for each band by calculating the mean of the difference of reconstructed pixel values and original pixel values. However, more coding bits are necessary. Statistical results show that the offsets of most pixel belong to a small domain. Thus in AVS2, only four bands are selected in order to save coding bits. Considering the fact that some sample values may be quite different with the others, 2 start band positions are transmitted to the decoder.

Besides EO and BO, merge technique is utilized in order to save the bits consuming, where a merge flag is employed to indicate whether the SAO parameters of the current LCU is exact the same with its neighbors. When merge flag is enabled, all the following SAO parameters are not signaled but inferred from neighbors.

ALF is the last stage of in-loop filtering. Its nature is to minimize the mean squared error between the original frame and the reconstructed frame using Wiener-Hopf equations. There are two stages in this process at encoder side. The first stage is filter coefficient derivation. To achieve the filter coefficients, reconstructed pixels of the luminance component are classified into 16 categories, and one set of filter coefficients is trained for each category using Wiener-Hopf equations. To reduce the redundancy between these 16 sets of filter coefficients, the encoder will adaptively merge them based on the rate-distortion performance. At its maximum, 16 different filter sets can be assigned for the luminance component and only one for each chrominance component. The second stage is to filter each sample with the corresponding derived filter coeffiencts using a 7x7 cross and 3x3 square filter as shown in **Fig. 12**.

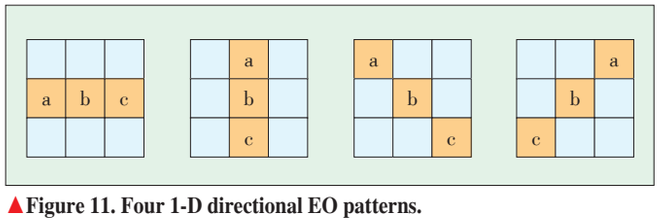
Finally, the filtered sample can be achieved as follows:

$$ptmp = C[filterIdx][8] \times p(x,y) + \sum_{j=0}^{7} C[filterIdx][j] \times (p(x - Hor[j], y - Ver[j]) + p(x + Hor[j], y + Ver[j])) \quad (6)$$

$$ptmp = (ptmp + 32) >> 6 \quad (7)$$

$$p'(x,y) = Clip3(0, (1 << BitDepth) - 1, ptmp) \quad (8)$$

where *filterIdx* indicates luma or chroma component, $p(x,y)$ is the reconstructed sample after SAO. $p'(x,y)$ is the final reconstructed sample after ALF. *Hor*[*j*]and *Ver*[*j*] stands for the filter coefficients positions.



▲**Figure 11. Four 1-D directional EO patterns.**

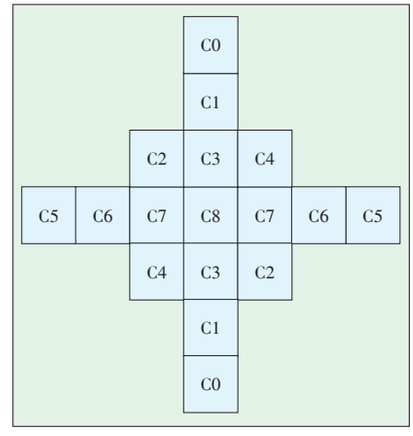
▼**Table 4. The classification rules and pixel categories**

Category	Condition	Offset Range
1	c < a && c < b	-1 <= offset <= 6
2	(c < a && c==b) || (c == a && c < b)	0 <= offset <= 1
3	(c > a && c==b) || (c == a && c > b)	-1 <= offset <= 0
4	c > a && c > b	-6 <= offset <= 1
0	None of the above	None

4 Scene Video Coding

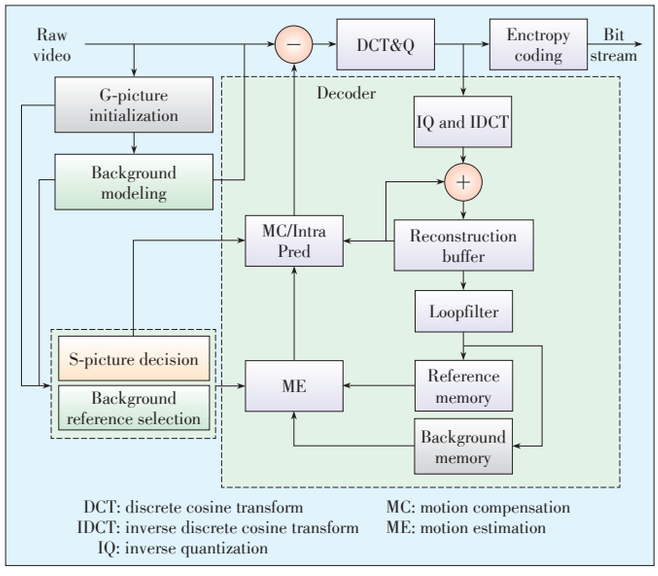
In practical applications, many videos are captured in specific scenes, such as surveillance video and videos from classroom, home, court, etc., which are characterized by temporally stable background. The redundancy originating from the background could be further reduced. In AVS2, a background-picture-model-based coding method is proposed to achieve higher compression performance [19] (**Fig. 13**). G-pictures and S-pictures are defined to further exploit the temporal redundancy and facilitate video event generation such as object segmentation and motion detection.

The G-picture is a special I-picture, which is stored in a separate background memory. It is encoded by intra mode only and is not decoded for displaying. The reason is that it is just for being referenced rather than for viewing. For the generation of a G-picture, a method of segment-and-weight based running



◀**Figure 12. Adaptive loop filter shape.**

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P7



▲Figure 13. Background picture based scene coding in AVS2.

average (SWRA) [20] is used to generate the GB picture. SWRA approximately generates the background by assigning larger weights on the frequent values in the averaging process. When encoding the G-picture, a smaller QP is selected to make a high-quality G-picture. Then the G-picture can be well referenced by the following pictures.

S-picture is a special P-picture that can only be predicted from a reconstructed G-picture or virtual G-picture which does not exist in the actual input sequence but is modeled from input pictures and encoded into the stream to act as a reference picture.Only intra, SKIP and P2N×2N modes with zero motion vectors are available in S picture. In the AVS2, the S picture is set as the random access point instead of intra-predicted picture. However, the S picture outperforms I picture when adopted as therandom access point since the inter prediction is adopted in theS picture and the prediction performance is better. With the S-picture, the performance of the random access can be improved on a large scale.

Furthermore, according to the predication modes in AVS2 compression bitstream, the blocks of an AVS2 picture could be classified as background blocks, foreground blocks or blocks on the edge area. Obviously, this information is very helpful for possible subsequent vision tasks, such as object detection and tracking. Object-based coding has already been proposed in MPEG-4; however, object segmentation remains a challenge that constrains the application of object based coding. Therefore, AVS2 uses simple background modeling instead of accurate object segmentation. The simple background modeling is easier and provides a good tradeoff between coding efficiency and complexity.

To provide convenience for applications like event detection and searching, AVS2 adds some novel high-level syntax to describe the region of interest (ROI). In the region extension, the region number, event ID, and coordinates for top left and bottom right corners are included to show what number the ROI is, what event happened and where it lies.

5 Performance Comparison

In this section, the performance comparisons among AVS2, AVS1, and state-of-the-art High Efficiency Video Coding (HEVC) international standard are provided. For comparison, the reference software used in the experiments is HM16.6 for HEVC, GDM 4.1 for AVS1 and RD12.0 for AVS2. HEVC and AVS1 are used as a testing anchor. According to the applications, we tested the performance of AVS2 with three different coding configurations: all-intra (AI), random access (RA), and low delay (LD), similar to the HEVC common test conditions and BD-Rate is used for bitrate saving evaluation. The UHD, 1080 p, 720 p, WVGA and WQVGA test sequences are the common test sequences used in AVS2, including partial test sequences used in HEVC, such as Traffic (UHD), Kimono1 (1080 p), BasketballPass (WQVGA) and City (720 p). Moreover, surveillance sequences including 1200 p and 576 p are tested to further compare the performance of AVS2 and HEVC under their respective common test condition. All these sequences and the surveillance/videoconference sequences are available on the AVS website.

Table 5 shows the rate distortion performance of AVS2 for three test cases. For different test configurations, AVS2 shows comparable performance as HEVC and outperforms AVS1 with significant bits saving, up to 52.9% for RA. **Table 6** shows the rate distortion performance comparisons of AVS2 with HEVC for surveillance sequences. AVS2 outperforms HEVC by

▼Table 5. Bitrate saving of AVS2 performance comparison with AVS1, HEVC for common test sequences

Sequences	AI configuration		RA configuration		LD configuration	
	AVS1 vs. AVS2	HEVC vs. AVS2	AVS1 vs. AVS2	HEVC vs. AVS2	AVS1 vs. AVS2	HEVC vs. AVS2
UHD	-31.2%	-2.21%	-50.5%	-0.29%	-57.6%	2.72%
1080 p	-33.1%	-0.67%	-51.3%	-2.30%	-44.3%	0.68%
720 p	-34.0%	-2.06%	-57.2%	-2.44%	-56.3%	1.88%
WVGA	-30.4%	1.46%	-52.8%	0.05%	-50.5%	0.91%
WQVGA	-26.6%	2.78%	-52.4%	1.08%	-49.4%	4.87%
Overall	-31.2%	-0.06%	-52.9%	-0.88%	-51.0%	2.11%

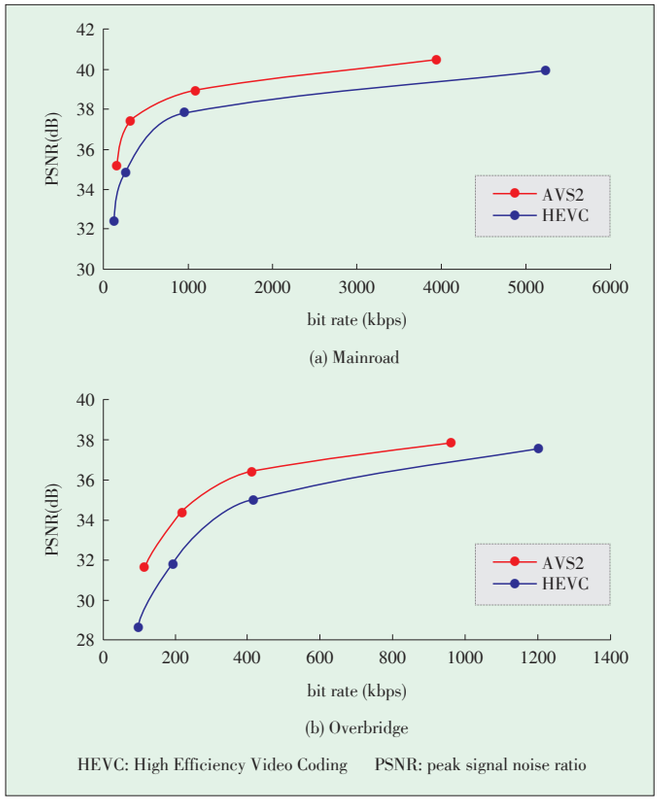
HEVC: High Efficiency Video Coding AI: all-intra LD: low delay RA: random access

▼Table 6. Bitrate saving of AVS2 performance comparison with HEVC for surveillance sequences

Sequences	RA configuration	LD configuration
1200 p	-35.7%	-38.5%
576 p	-41.3%	-26.5%
Overall	-39.1%	-31.3%

LD: low delay RA: random access

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P8



▲**Figure 14. Performance comparison between AVS2 and HEVC for surveillance videos: (a) MainRoad, (b) Overbridge.**

39.1% and 31.3% under RA and LD test configuration, respectively. The curves in **Fig. 14** show the results on two surveillance video sequences.

6 Conclusions

This paper gives an overview of the AVS2 standard. AVS2 is an application oriented coding standard, and different coding tools have been developed according to various application characteristics and requirements. For high quality broadcasting, flexible prediction and transform coding tools have been incorporated. Especially for scene applications, AVS2 significantly improves coding performance and bridges video compression with machine vision by incorporating the background picture modeling, thereby making video coding smarter and more efficient. In a word, compared to the previous AVS1 coding standard, AVS2 achieves significant improvement both in coding efficiency and flexibility.

References

[1] S. Ma, S. Wang, and W. Gao, "Overview of IEEE 1857 video coding standard," in *Proc. IEEE International Conference on Image Processing*, Melbourne, Australia, Sept. 2013, pp.1500–1504. doi: 10.1109/MSP.2014.2371951.
[2] Q. Yu, S. Ma, Z. He, *et al.*, "Suggested video platform for AVS2," 42nd AVS Meeting, Guilin, China, AVS_M2972, Sept. 2012.
[3] Y. Piao, S. Lee and C. Kim, "Modified intra mode coding and angle adjustment," 48th AVS Meeting, Beijing, China, AVS_M3304, Apr. 2014.
[4] Q. Yu, X. Cao, W. Li, *et al.*, "Short distance intra prediction," 46th AVS Meeting, Shenyang, China, AVS_M3171, Sept. 2013.
[5] Y. Piao, S. Lee, I.-K. Kim, and C. Kim, "Derived mode (DM) for chroma intra prediction," 44th AVS Meeting, Luoyang, China, AVS_M3042, Mar. 2013.
[6] Y. Lin and L. Yu, "F frame CE: Multi forward hypothesis prediction," 48th AVS Meeting, Beijing, China, AVS_M3326, Apr. 2014.
[7] Z. Shao and L. Yu, "Multi-hypothesis skip/direct mode in P frame," 47th AVS Meeting, Shenzhen, China, AVS_M3256, Dec. 2013.
[8] Y. Ling, X. Zhu, L. Yu, *et al.*, "Multi-hypothesis mode for AVS2," 47th AVS meeting, Shenzhen, China, AVS_M3271, Dec. 2013.
[9] I.-K. Kim, S. Lee, Y. Piao, and C. Kim, "Directional multi-hypothesis prediction (DMH) for AVS2," 45th AVS Meeting, Taicang, China, AVS_M3094, Jun. 2013.
[10] H. Lv, R. Wang, Z. Wang, *et al.*, "Sequence level adaptive interpolation filter for motion compensation," 47th AVS Meeting, Shenzhen, China, AVS_M3253, Dec. 2013.
[11] Z. Wang, H. Lv, X. Li, *et al.*, "Interpolation improvement for chroma motion compensation," 48th AVS Meeting, Beijing, China, AVS_M3348, Apr. 2014.
[12] J. Ma, S. Ma, J. An, K. Zhang, and S. Lei, "Progressive motion vector precision," 44th AVS Meeting, Luoyang, China, AVS_M3049, Mar. 2013.
[13] S. Lee, I.-K. Kim, Min-Su Cheon, N. Shlyakhov, and Y. Piao, "Proposal for AVS2.0 Reference Software," 42nd AVS Meeting, Guilin, China, AVS_M2973, Sept. 2012.
[14] W. Li, Y. Yuan, X. Cao, et al., "Non-square quad-tree transform," 45th AVS Meeting, Taicang, China, AVS_M3153, Jun. 2013.
[15] J. Wang, X. Wang, T. Ji, and D. He, "Two-level transform coefficient coding," 43rd AVS Meeting, Beijing, China, AVS_M3035, Dec. 2012.
[16] X. Wang, J. Wang, T. Ji, and D. He, "Intra prediction mode based context design," 45th AVS Meeting, Taicang, China, AVS_M3103, Jun. 2013.
[17] J. Chen, S. Lee, C. Kim, *et al.*, "Sample adaptive offset for AVS2," 46th AVS Meeting, Shenyang, China, AVS_M3197, Sept. 2013.
[18] X. Zhang, J. Si, S. Wang, *et al.*, "Adaptive loop filter for AVS2," 48th AVS Meeting, Beijing, China, AVS_M3292, Apr. 2014.
[19] S. Dong, L. Zhao, P. Xing, and X. Zhang, "Surveillance video coding platform for AVS2," 47th AVS Meeting, Shenzhen, China, AVS_M3221, Dec. 2013.
[20] X. Zhang, Y. Tian, T. Huang, and W. Gao, "Low-complexity and high efficiency background modelling for surveillance video coding," in *IEEE International Conference on Visual Communication and Image Processing*, San Diego, USA, Nov. 2012, pp. 1–6. doi: 10.1109/VCIP.2012.6410796.

Manuscript received: 2015-11-16

Biographies

Shanshe Wang (sswang@pku.edu.cn) received the BS degree in Department of Mathematics from Heilongjiang University, China in 2004, MS degree in computer software and theory from Northeast Petroleum University, China in 2010, and PhD degree in computer science from the Harbin Institute of Technology, China. Now he is a post doctor of Computer Science, National Engineering Lab. on Video Technology, Peking University, China. His current research interests include video compression and image and video quality assessment.

Falei Luo (falei.luo@vipl.ict.ac.cn) received the BS degree from Huazhong University of Science and Technology, China and is currently pursuing the PhD degree at Institute of Computing Technology, Chinese Academy of Sciences, China.

Siwei Ma (swma@pku.edu.cn) received the BS degree from Shandong Normal University, China in 1999, and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China in 2005. From 2005 to 2007, he held a post-doctorate position with the University of Southern California, Los Angeles, USA. Then, he joined the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, China, where he is currently a professor of Computer Science, National Engineering Lab. on Video Technology, and a co-chair of AVS video Subgroup. He has published over 100 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

D:\EMAG\2016-02-49/VOL13\F4.VFT——9PPS/P9