Siwei Ma, Tiejun Huang, Cliff Reader, and Wen Gao

AVS2—Making Video Coding Smarter

AVS2 is a new generation of video coding standard developed by the IEEE 1857 Working Group under project 1857.4. AVS2 is also the second-generation video coding standard established by the Audio and Video Coding Standard (AVS) Working Group of China; the first-generation AVS1 was developed by the AVS Working Group and issued as Chinese national standard GB/T 20090.2-2006 in 2006. The AVS Working Group was founded in 2002 and is dedicated to providing the digital audio-video industry with highly efficient and economical coding/decoding technologies. So far, the AVS1 video coding standard is widely implemented in regional broadcasting, communication, and digital video entertainment systems. As the successor of AVS1, AVS2 is designed to achieve significant coding efficiency improvements relative to the preceding H.264/MPEG4-AVC and AVS1 standards. The basic coding framework of AVS2 is similar to the conterminous HEVC/H.265, but AVS2 can provide more efficient compression for certain video applications such as surveillance as well as low-delay communication such as videoconferencing. AVS2 is making video coding smarter by adopting intelligent coding tools that not only improve coding efficiency but also help with computer vision tasks such as object detection and tracking.

BACKGROUND

The AVS Working Group was established in March 2002 in China. The mandate of the group is to establish generic technical standards for the compression, decoding, processing, and representation of digital audio-video content, thereby enabling digital audio-video equipment and systems with highly efficient and economical coding/decoding technologies. After more than a decade, the working group has published a series of standards, including AVS1, which is the culmination of the first stage of work.

AVS2 IS MAKING VIDEO CODING SMARTER BY ADOPTING INTELLIGENT CODING TOOLS THAT NOT ONLY IMPROVE CODING EFFICIENCY, BUT ALSO HELP WITH COMPUTER VISION TASKS SUCH AS OBJECT DETECTION AND TRACKING.

Table 1 shows the time line of the AVS1 video coding standard (for short, AVS1). In AVS1, six profiles were defined to meet the requirements of various applications. The Main Profile focuses on digital video applications like commercial broadcasting and storage media, including high-definition video applications. It was approved as a national standard in China: GB/T 20090.2-2006. It was followed by the Enhanced Profile, an extension of the Main Profile with higher coding efficiency, targeting the needs of multimedia entertainment, such as movie compression for high-density storage. The Surveillance Baseline and Surveillance Profiles focus on video surveillance applications, considering in particular the characteristics of surveillance videos, i.e., high noise levels, relatively low encoding complexity, and requirements for easy event detection and search. The Portable Profile targets mobile video applications with lower resolution, low computational complexity, and robust error resiliency to meet the wireless environment. The latest Broadcasting Profile is also an improvement of the Main Profile and targets high-quality, high-definition TV (HDTV) broadcasting. It was approved and published as an industry standard by the State of China Broadcasting Film and Television Administration in July 2012.

AVS standards are also being recognized internationally. In 2007, the Main

[TABLE 1] TIME LINE OF AVS1 VIDEO CODING STANDARD.

TIME	PROFILE	TARGET APPLICATION(s)	MAJOR CODING TOOLS
DECEMBER 2003	MAIN	TV BROADCASTING	8 × 8 BLOCK-BASED INTRAPREDICTION, TRANSFORM AND DEBLOCKING FILTER; VARIABLE BLOCK SIZE MOTION COMPENSATION (16 × 16 ~ 8 × 8)
JUNE 2008	SURVEILLANCE BASELINE	VIDEO SURVEILLANCE	BACKGROUND-PREDICTIVE PICTURE FOR VIDEO CODING, ADAPTIVE WEIGHTING QUANTIZATION (AWQ), CORE FRAME CODING
SEPTEMBER 2008	ENHANCED	DIGITAL CINEMA	CONTEXT BINARY ARITHMETIC CODING (CBAC), AWQ
JULY 2009	PORTABLE	MOBILE VIDEO COMMUNICATION	8 × 8/4 × 4 BLOCK TRANSFORM
JULY 2011	SURVEILLANCE	VIDEO SURVEILLANCE	BACKGROUND MODELING BASED CODING
MAY 2012	BROADCASTING	HDTV	AWQ, ENHANCED FIELD CODING

Digital Object Identifier 10.1109/MSP.2014.2371951
Date of publication: 12 February 2015

1053-5888/15©2015IEEE

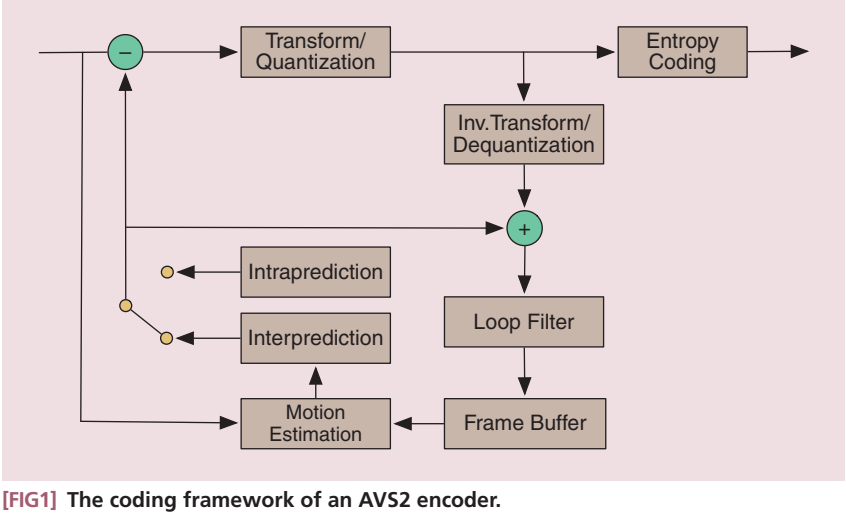
Profile was accepted as an option of video codecs for Internet Protocol Television (IPTV) applications by the International Telecommunication Union–Telecommunication Standardization Sector (ITU-T) Focus Group on IPTV standardization [1]. The IEEE 1857 Working Group was established in 2012 to work on IEEE standards for advanced audio and video coding, based on individual members of the IEEE Standards Association from the AVS Working Group. The IEEE 1857 Working Group meets three to four times annually to discuss the standard technologies, syntax, and so on. Until now, the IEEE 1857 Working Group has finished three parts of IEEE 1857 standards, including IEEE 1857-2013 for video, IEEE 1857.2-2013 for audio, and IEEE 1857.3-2013 for system [2].

AVS standards have been developed in compliance with the AVS intellectual property rights (IPR) policy. This policy includes up-front commitment by participants to license essential patents with declaration of default licensing terms—royalty-free without compensation [(RAND-RF) and otherwise under reasonable and nondiscriminatory terms], or participation in the AVS patent pool, or RAND. The disclosure of published patent applications and granted patents is required, and the existence of unpublished applications is also required if the RAND option is taken. The licensing terms are also considered in the adoption of proposals for AVS standards when all technical factors are equal. Reciprocity in licensing is required. The protection of participants's IPR is provided to guard against the situation in which the IPR of a participant are disclosed by another party. AVS has encouraged the establishment of a Patent Pool Administration (PPA) that is independent from the AVS Working Group, which only focuses on the standards. The AVS standards are also fully compliant with the IPR policy of IEEE standards.

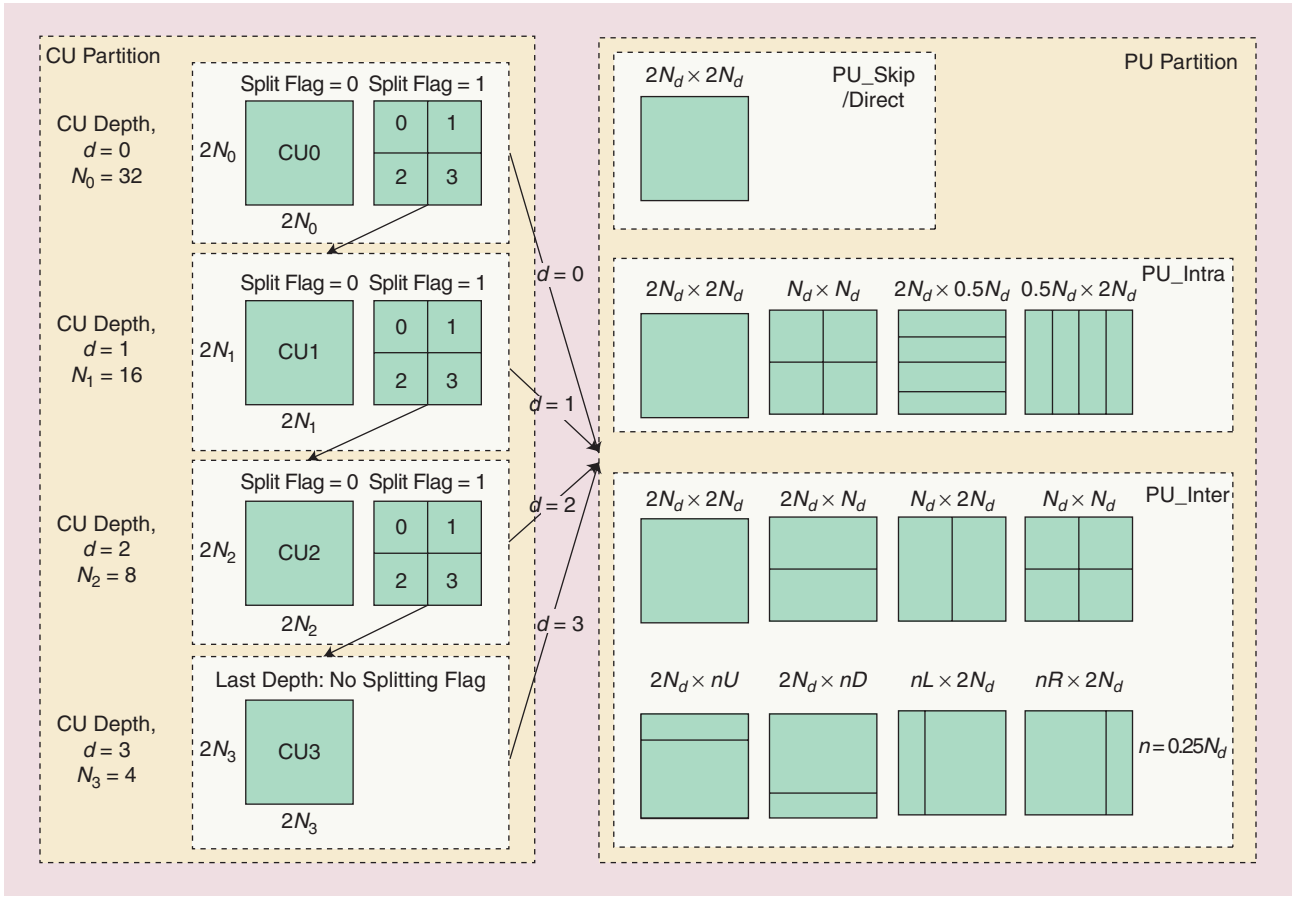
AS A SUCCESSOR OF AVS1, AVS2 IS DESIGNED TO IMPROVE CODING EFFICIENCY FOR HIGHER RESOLUTION VIDEOS AND PROVIDE EFFICIENT COMPRESSION SOLUTIONS FOR VARIOUS KINDS OF VIDEO APPLICATIONS.

Based on the success of AVS1 and the recent research and standardization works, AVS has been working on a new generation of video coding technologies called AVS2 (or more specifically, Part 2 in the AVS2 series standards). In fact, since 2005 and before the AVS2 project officially started, AVS has been continuously working on an AVS-X project to explore more efficient coding techniques. AVS2 was started formally by issuing a call for platforms in March 2012. By October 2012, a reference platform (RD 1.0) based on the AVS1 reference software was developed for AVS2 [3]. After that, AVS2 continued to improve its coding efficiency, and the standard in committee draft 2.0 was finalized in June 2014. It has been approved as a project of IEEE standard, IEEE 1857.4, and a project of Chinese national standard, both of which are expected to be finished by the end of 2014 at the time of this writing.

As a successor of AVS1, AVS2 is designed to improve coding efficiency for higher-resolution videos and provide efficient compression solutions for various kinds of video applications. Compared to the preceding coding standards, AVS2 adopts smarter coding tools that are adapted to satisfy the new requirements identified from emerging applications. First, more flexible prediction block partitions are used to further improve prediction accuracy, e.g., square and nonsquare partitions, which are more adaptive to the image content especially in edge areas. Related to the prediction structure, transform block size is more flexible and can be up to 64 × 64 pixels. After transformation, context adaptive arithmetic coding is used for the entropy coding of the transformed coefficients. A two-level coefficient scan and coding method can encode the coefficients of large blocks more efficiently. Moreover, for low-delay communication applications, e.g., video surveillance, video conference, etc., where the background usually does not often change, a background picture model-based coding method is developed in AVS2. The background picture constructed from original pictures or decoded pictures is used as a reference picture to improve prediction efficiency. Test results show that this background picture-based prediction coding can improve coding efficiency significantly. Furthermore, the background picture can also be used for object detection and tracking for intelligent surveillance. In addition, to support object tracking among multiple cameras in surveillance applications, navigation information such as those from the global positioning system and BeiDou Navigation Satellite System of China is also defined, which mainly includes timing, location, and movement information. Finally, aiming at more intelligent surveillance video coding, AVS2 also started a



[FIG1] The coding framework of an AVS2 encoder.



[FIG2] (a) The maximum possible recursive CU structure in AVS2. (LCU size = 64, maximum hierarchical depth = 4). (b) Possible PU splitting for skip, intramodes, and intermodes in AVS2, including symmetric and asymmetric prediction (*d*=1, 2 for intraprediction, and *d*= 0,1,2 for interprediction).

digital media content description project in which visual objects in the images or videos are described with multilevel features for facilitating visual object based storage, retrieval, and interactive applications, etc.

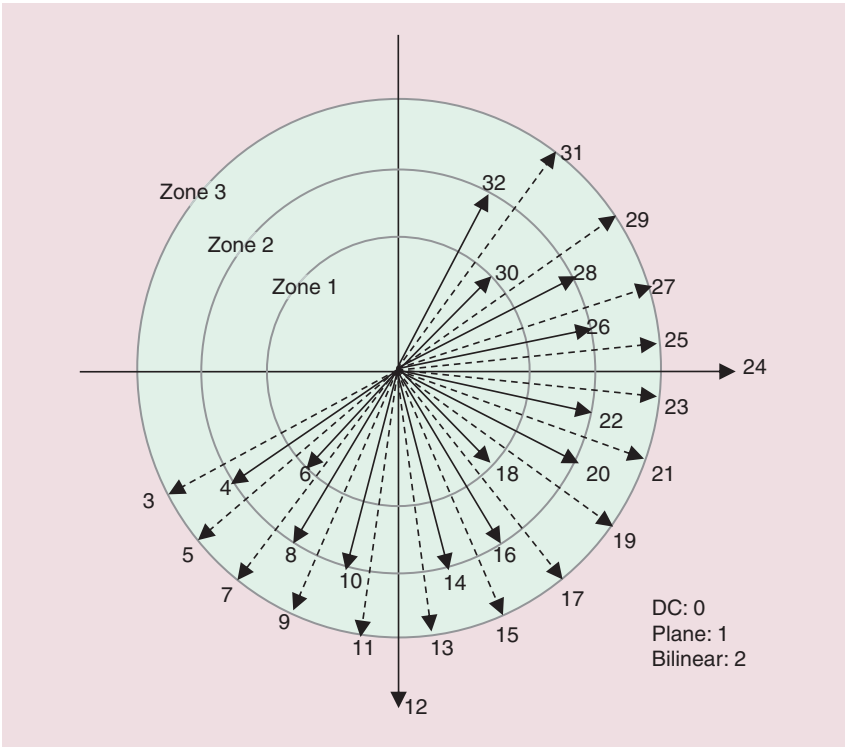
This column will provide a short overview of AVS2 video coding technology and a performance comparison with other video coding standards.

TECHNOLOGY AND KEY FEATURES

Similar to previous coding standards, AVS2 adopts the traditional prediction/transform hybrid coding framework, as shown in Figure 1. Within the framework, a more flexible coding structure is adopted for efficient high-resolution video coding, and more efficient coding tools are developed to make full use of the textual information and temporal redundancies. These tools can be classified into four categories: 1) prediction coding (including intraprediction and interprediction), 2) transform, 3) entropy coding, and 4) in-loop filtering. We will give a brief introduction to the coding framework and coding tools.

CODING FRAMEWORK

In AVS2, a coding unit (CU)-, prediction unit (PU)-, and transform unit (TU)-based coding/prediction/transform structure is adopted to represent and organize the encoded data [3]. First, pictures are split into largest coding units (LCUs), which consist of $2N \times 2N$ samples of a 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 limit, as shown in Figure 2(a). Once the splitting of the CU hierarchical tree is finished, the leaf node CUs can be further split into PUs. PU is the basic unit for intra- and interprediction and allows multiple different shapes to encode irregular image patterns, as shown in Figure 2(b). The size of a PU is limited to that of a CU with various square or rectangular shapes. More specifically, both intra- and interprediction partitions can be symmetric or asymmetric. Intraprediction partitions vary in the set $\{2N \times 2N, N \times N, 2N \times 0.5N, 0.5N \times 2N\}$, while inter-prediction partitions vary in the set $\{2N \times 2N, 2N \times N, N \times 2N, 2N \times nU, 2N \times nD, nL \times 2N, nR \times 2N\}$, where U, D, L,and R are the abbreviations of "Up," "Down," "Left," and "Right," respectively. 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.



[FIG3] **An illustration of directional prediction modes.**

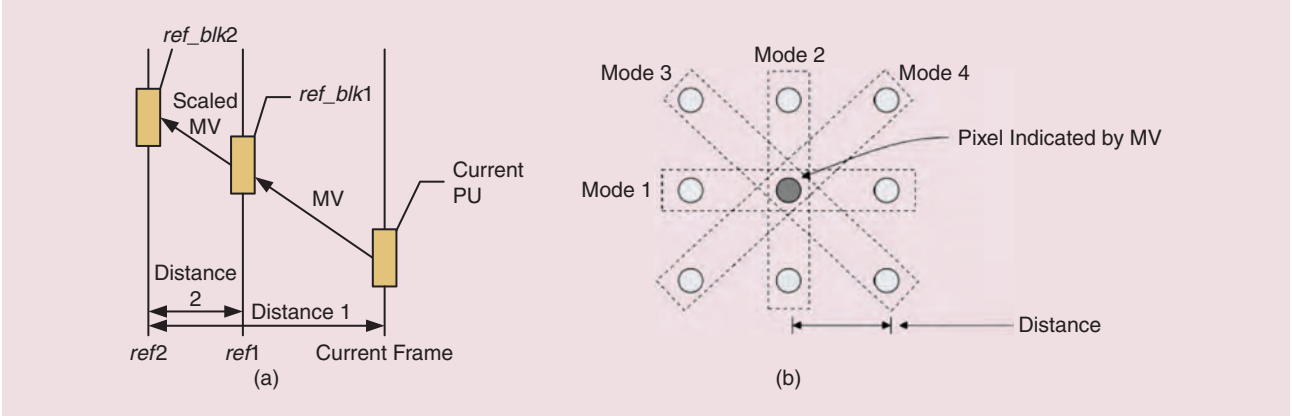
INTRAPREDICTION

Intraprediction is used to reduce the redundancy existing in the spatial domain of the picture. Block partition-based directional prediction is used for AVS2 [5]. As shown in Figure 2, besides the square PU partitions, nonsquare partitions, called *short distance intra prediction* (*SDIP*), are adopted by AVS2 for more efficient intraluminance prediction [4], where the nearest reconstructed boundary pixels are used as the reference sample in intraprediction. For SDIP, a $2N \times 2N$ PU is horizontally/vertically partitioned into four prediction blocks. SDIP is more adaptive to the image content, especially in edge area. But for the complexity reduction, SDIP is used in all CU sizes except a 64 × 64 CU. For each prediction block in the partition modes, a total of 33 prediction modes are supported for luminance, including 30 angular modes [5], a plane mode, a bilinear mode, and a DC mode. Figure 3 shows the distribution of the prediction directions associated with the 30 angular modes. Each sample in a PU is predicted by projecting its location to the reference pixels applying the selected prediction direction. To improve the intraprediction accuracy, the subpixel precision reference samples must be interpolated if the projected reference samples locate on a noninteger position. The noninteger position is bounded to 1/32 sample precision to avoid floating point operation, and a four-tap linear interpolation filter is used to get the subpixel.

For the chrominance component, the PU size is always $N \times N$, and five prediction modes are supported, including vertical prediction, horizontal prediction, bilinear prediction, DC prediction, and the prediction mode derived from the corresponding luminance prediction mode [6].

INTERPREDICTION

Compared to the spatial intraprediction, interprediction focuses on exploiting the temporal correlation between the consecutive pictures to reduce the temporal redundancy. Multireference prediction has been used since the H.264/AVC standard, including both short-term and long-term reference pictures. In AVS2, long-term reference picture usage is extended further, which can be constructed from a sequence of long-term decoded pictures, e.g., background picture used in surveillance coding, which will be discussed separately later. For short-term reference prediction in AVS2, F frames are defined as a special P frame [7], in addition to the traditional P and B frames. More specifically, a P frame is a forward-predicted frame using a single reference picture, while a B frame is a bipredicted frame that consists of forward,



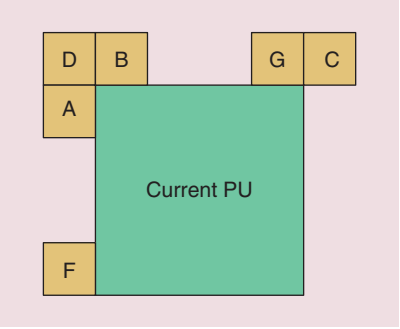
[FIG4] **(a) Temporal multihypothesis mode. (b) Spatial multihypothesis mode.**

backward, biprediction, and symmetric prediction, using two reference frames.

In a B frame, in addition to the conventional forward, backward, bidirectional, and skip/direct prediction modes, symmetric prediction is defined as a special biprediction mode, wherein only one forward motion vector (MV) is coded and the backward MV is derived from the forward MV. For an F frame, besides the conventional single hypothesis prediction mode in a P frame, multihypothesis techniques are added for more efficient prediction, including the advanced skip/direct mode [8], temporal multihypothesis prediction mode [9], and spatial directional multihypothesis (DMH) prediction mode [10].

In an F frame, an advanced skip/direct mode is defined using a competitive motion derivation mechanism. Two derivation methods are used: one is temporal and the other is spatial. Temporal multihypothesis mode combines two predictors along the predefined temporal direction, while spatial multihypothesis mode combines two predictors along the predefined spatial direction. For temporal derivation, the prediction block is obtained by an average of the prediction blocks indicated by the MV prediction (MVP) and the scaled MV in a second reference. The second reference is specified by the reference index transmitted in the bit stream. For temporal multihypothesis prediction, as shown in Figure 4, one predictor *ref_blk*1 is generated with the best MV *MV* and a reference frame *ref*1 is searched by motion estimation, and then this MV is linearly scaled to a second reference to generate another predictor *ref_blk*2. The second reference *ref*2 is specified by the reference index transmitted in the bit stream. In DMH mode, as specified in Figure 4, the seed predictors are located on the line crossing the initial predictor obtained from motion estimation. The number of seed predictors is restricted to eight. If one seed predictor is selected for combined prediction, for example "Mode 1," then the index of the seed predictor "1" will be signaled in the bit stream.

For spatial derivation, the prediction block may be obtained from one or two prediction blocks specified by the motion copied from its spatial neighboring blocks. The neighboring blocks are illustrated in Figure 5. They are searched in a predefined order F, G, C, A, B, D, and the selected neighboring block is signaled in the bit stream.



[FIG5] An illustration of neighboring blocks A, B, C, D, F, and G for MVP.

MOTION VECTOR PREDICTION AND CODING

MVP plays an important role in interprediction, which can reduce the redundancy among MVs of neighboring blocks and thus save large numbers of coding bits for MVs. In AVS2, four different prediction methods are adopted, as tabulated in Table 2. Each of them has its unique usage. Spatial MVP is used for the spatial derivation of Skip/Direct mode in F frames and B frames. Temporal MVP is used for temporal derivation of Skip/Direct mode in P frames and F frames. Spatial-temporal-combined MVP is used for the joint temporal and spatial derivation of Skip/Direct mode in B frames. For other cases, median prediction is used.

In AVS2, the MV is in quarter-pixel precision for the luminance component, and the subpixel is interpolated with an eight-tap DCT interpolation filter (DCT-IF) [11]. For the chrominance component, the MV derived from luminance with 1/8 pixel precision and a four-tap DCT-IF is used for subpixel interpolation [12]. After the MVP, the MV difference (MVD) is coded in the bit stream. However, redundancy may still exist in MVD, and to further save coding bits of MVs, a progressive MV resolution adaptation method is adopted in AVS2 [13]. In this scheme, the MVP is firstly rounded to the nearest integer sample position, and then the MV is rounded to a half-pixel precision if its distance from MVP is larger than a by a threshold. Finally, the resolution of the MVD is decreased to half-pixel precision if it is larger than a threshold.

TRANSFORM

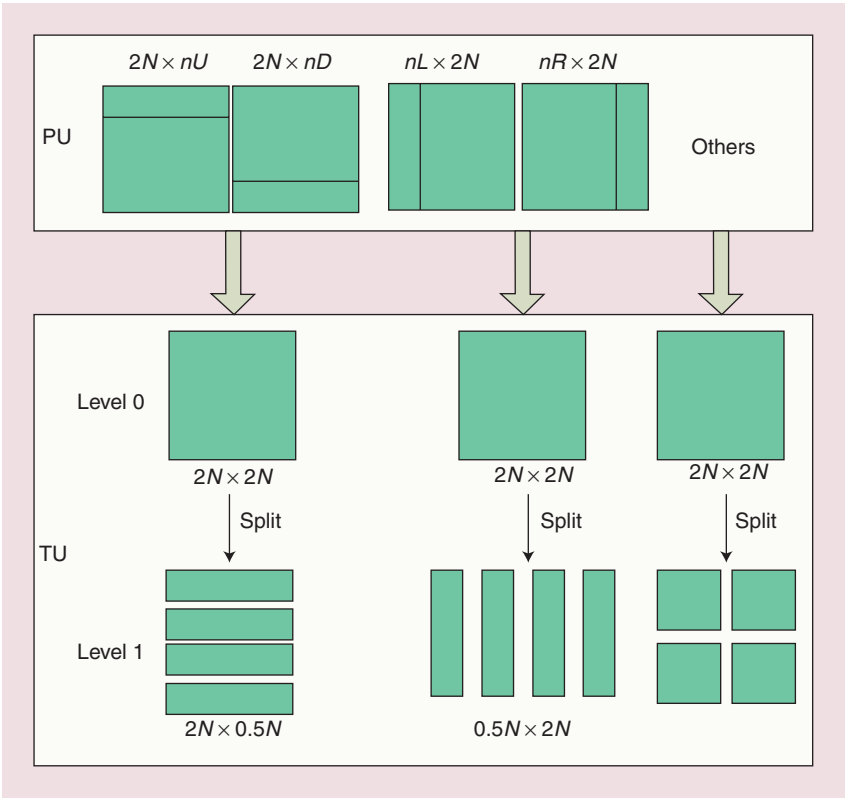
Two-level transform coding is utilized to further compress the predicted residual. For a CU with symmetric prediction unit partition, the TU size can be $2N \times 2N$ or $N \times N$ signaled by a transform split flag. Thus, the maximum transform size is 64 × 64, and the minimum transform size is 4 × 4. For the TU size 4 × 4 to 32 × 32, an integer transform (IT) that closely approximates the performance of the discrete cosine transform (DCT) is used; while for the 64 × 64 transform, a logical transform (LOT) [14] is applied to the residual. A five-three-tap integer wavelet transform is first performed on a 64 × 64 block discarding the low-high (LH), high-low (HL), and (high-high) HH-bands, and then a normal 32 × 32 IT is applied to the low-low (LL)-band. For a CU that has an asymmetric PU partition, a $2N \times 2N$ IT is used in the first level and a nonsquare transform [15] is used in the second level, as shown in Figure 6. Moreover, in the latest AVS2 standard, a secondary transform was adopted for intraprediction residual (for more details see the latest AVS specification document N2120 on the AVS FTP Web site [21]).

ENTROPY CODING

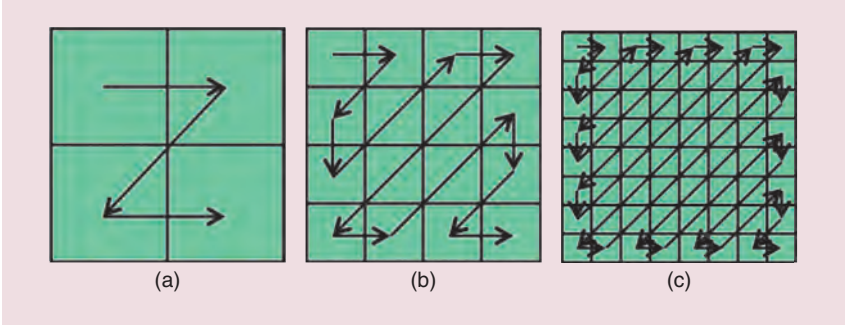
After transform and quantization, a two-level coding scheme is applied to the

[TABLE 2] MV PREDICTION METHODS IN AVS2.

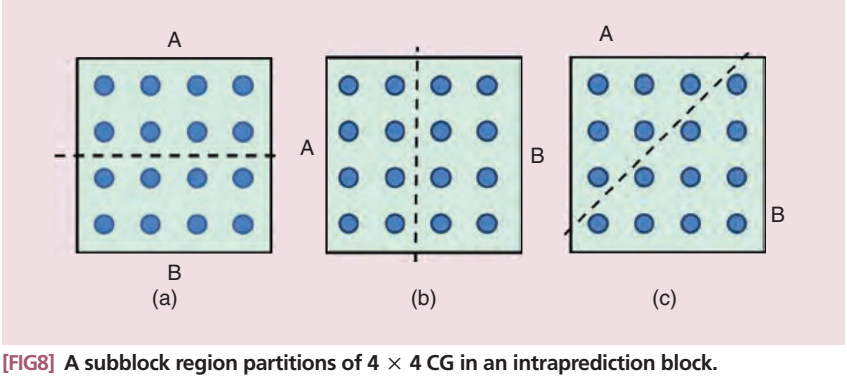
METHOD	DETAILS
MEDIAN	USING THE MEDIAN MV VALUES OF THE NEIGHBORING BLOCKS.
SPATIAL	USING THE MVs OF SPATIAL NEIGHBORING BLOCKS.
TEMPORAL	USING THE MVs OF TEMPORAL COLLOCATED BLOCKS.
SPATIAL-TEMPORAL COMBINED	USING THE TEMPORAL MVP FIRST IF IT IS AVAILABLE, AND SPATIAL MVP IS USED INSTEAD IF THE TEMPORAL MVP IS NOT AVAILABLE.



[FIG6] A PU partition and two-level transform coding.



[FIG7] A subblock scan for transform blocks of size (a) 8 × 8, (b) 16 × 16, and (c) 32 × 32 transform blocks; each subblock represents a 4 × 4 CG.



[FIG8] A subblock region partitions of 4 × 4 CG in an intraprediction block.

transform coefficient blocks [16]. A coefficient block is partitioned into 4 × 4 coefficient groups (CGs), as shown in Figure 7. Then zig-zag scanning and context-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 nonzero CG is coded first, followed by a bin string of significant CG flags indicating whether the CG scanned in zig-zag order contains nonzero coefficients. At the coefficient level, for each nonzero CG, the coefficients are further scanned into the form of (*run*, *level*) pair in zig-zag order. Level and run refer to the magnitude of a nonzero coefficient and the number of zero coefficients between two nonzero coefficients, respectively. For the last CG, the coefficient position that denotes the position of the last nonzero coefficient in scan order is coded first. For a nonlast CG, a last run is coded that denotes number of zero coefficients after the last nonzero coefficient in zig-zag scan order. And then the (*level*, *run*) pairs in a CG are coded in reverse zig-zag scan order.

For the context modeling used in the CABAC, AVS2 employs a mode-dependent context selection design for intraprediction blocks [17]. In this context design, 34 intraprediction 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, as shown in Figure 8. The intraprediction modes and CG regions are applied in the context coding of syntax elements including the last CG position, last coefficient position, and run value.

IN-LOOP FILTERING

Artifacts such as blocking artifacts, ringing artifacts, color biases, and blurring artifacts are quite common in compressed video, especially at medium and low bit rate. To suppress those artifacts, deblocking filtering, sample adaptive offset (SAO) filtering [18], and adaptive loop filter (ALF) [19] are applied to the reconstructed pictures sequentially.

Deblocking filtering aims to remove the blocking artifacts caused by block transform and quantization. The basic unit for the deblocking filter is an 8×8 block. For each 8×8 block, the deblocking filter is used only if the boundary belongs to either of the CU, PU, or TU boundaries.

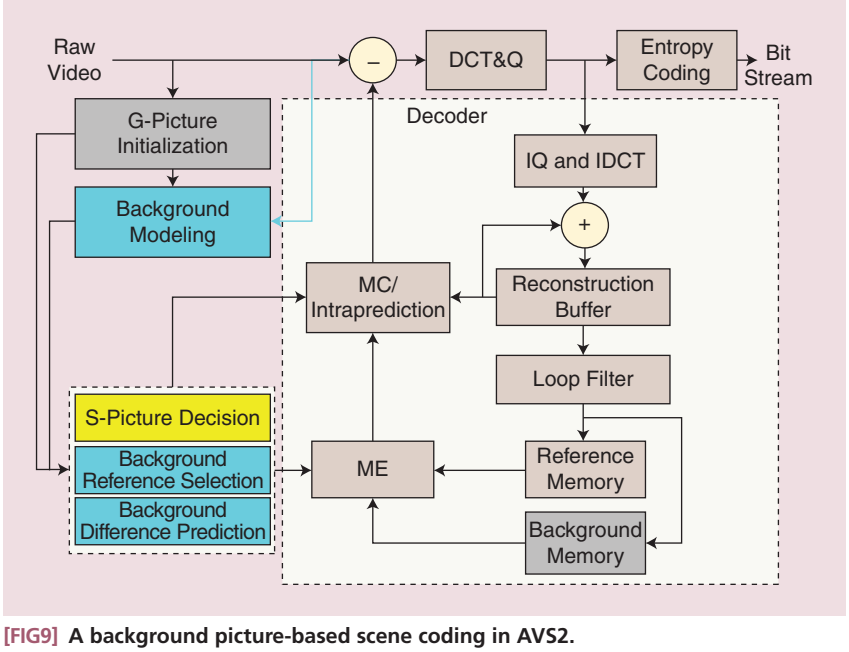
After the deblocking filter, an SAO filter is applied to reduce the mean sample distortion of a region, where an offset is added to the reconstructed sample to reduce ringing artifacts and contouring artifacts. There are two kinds of offset: edge offset (EO) and band offset (BO) mode. For the EO mode, the encoder can select and signal a vertical, horizontal, downward-diagonal, or upward-diagonal filtering direction. For BO mode, an offset value that directly depends on the amplitudes of the reconstructed samples is added to the reconstructed samples.

ALF is the last stage of in-loop filtering. There are two stages in this process. The first stage is filter coefficient derivation. To train the filter coefficients, the encoder classifies reconstructed pixels of the luminance component into 16 categories, and one set of filter coefficients is trained for each category using Wiener–Hopf equations to minimize the mean squared error between the original frame and the reconstructed frame. 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 the chrominance components. The second stage is a filter decision, which includes both the frame level and LCU level. First, the encoder decides whether frame-level adaptive loop filtering is performed. If frame level ALF is on, then the encoder further decides whether the LCU level ALF is performed.

SMART SCENE VIDEO CODING

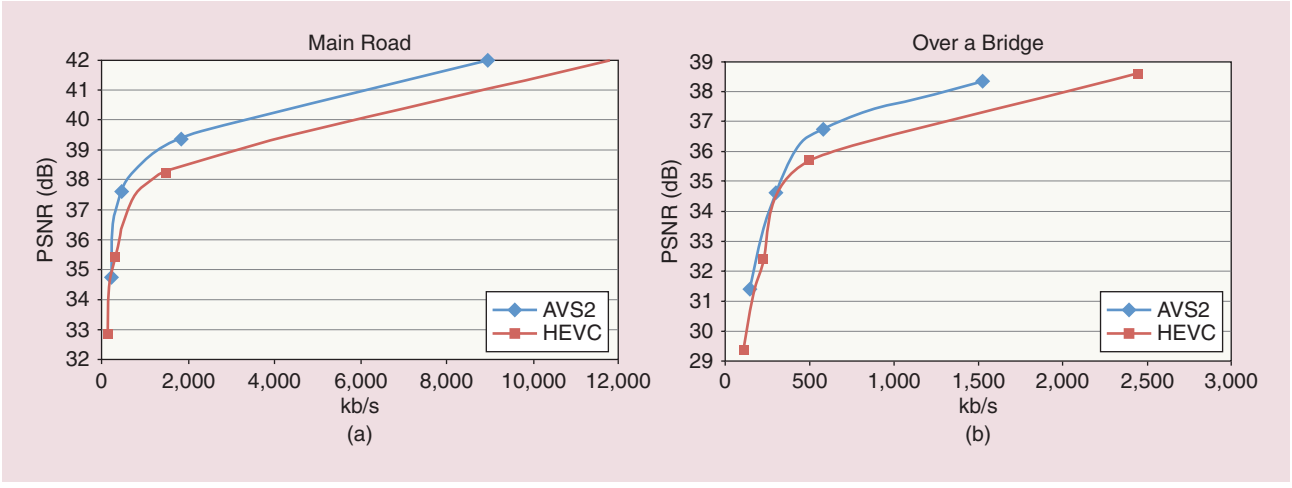
More and more videos being captured in specific scenes (such as surveillance video and videos from the classroom, home, courthouse, etc.) are characterized by a temporally stable background. The redundancy originating from the background could be further reduced. AVS2 developed a background picture model-based coding method [20], which is illustrated in Figure 9. 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. The S-picture is a special P-picture, which can be only predicted from a reconstructed G-picture or a 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.

The G-picture is initialized by background initialization and updated by background modeling with methods such as median filtering, fast implementation



[FIG9] **A background picture-based scene coding in AVS2.**

[FIG10] **Examples of the background picture and the difference frame between the original picture and the background picture: (a) original picture, (b) difference frame, and (c) background picture.**



[FIG11] A performance comparison between AVS2 and HEVC for surveillance videos: (a) main road and (b) over a bridge.

of a Gaussian mixture model, etc. In this way, the selected or generated G-picture can well represent the background of a scene with rare occluding foreground objects and noise. Once a G-picture is obtained, it is encoded and the reconstructed picture is stored into the background memory in the encoder/decoder and updated only if a new G-picture is selected or generated. After that, S-pictures can be involved in the encoding process by an S-picture decision. Except that it uses a G-picture as a reference, the S-picture owns similar properties as the traditional I-picture such as error resilience and random access (RA). Therefore, the pictures that should be coded as traditional I-pictures can be candidate S-pictures, such as the first picture of one group of pictures, or scene change, etc. Besides bringing about more prediction opportunity for those background blocks that normally dominate a picture, an additional benefit from the background picture is a new prediction mode called *background difference prediction*, as shown in Figure 10, which can improve foreground prediction performance by excluding the background influence. It can be seen that, after background difference prediction, the background redundancy is effectively removed. Furthermore, according to the predication modes in the AVS2 compression bit stream, 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 challenging problem, which constrains the application of object-based coding. Therefore AVS2 uses simple background modeling instead of accurate object segmentation, which is easier and provides a good tradeoff between coding efficiency and complexity.

AVS2 HAS BEEN DEVELOPED IN ACCORDANCE WITH AVS AND IEEE IPR POLICIES TO ENSURE RAPID LICENSING OF ESSENTIAL PATENTS AT COMPETITIVE ROYALTY RATES.

To provide convenience for applications like event detection and searching, AVS2 added 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.

PERFORMANCE COMPARISON

The major target applications of AVS2 are high-quality TV broadcasting and scene videos. For high-quality broadcasting, RA is necessary and may be achieved by inserting intraframes at a fixed interval, e.g, 0.5 s. And for high-quality video capture and editing, all intracoding (AI) is required. For scene video applications, e.g., video surveillance or videoconference, low delay (LD) needs to be guaranteed. According to the applications, we tested

[TABLE 3] BIT RATE SAVING OF AVS2 PERFORMANCE COMPARISON WITH AVS1 AND HEVC.

SEQUENCES	AI CONFIGURATION		RA CONFIGURATION		LD CONFIGURATION
	AVS2 VERSUS AVS1	AVS2 VERSUS HEVC	AVS2 VERSUS AVS1	AVS2 VERSUS HEVC	AVS2 VERSUS HEVC
UHD	31.2%	2.4%	50.3%	–0.4%	
1080P	33%	0.8%	50.3%	0.3%	
1200P					37.9%
SD					26.2%
OVERALL	32.1%	1.6%	50.3%	–0.1%	32.1%

the performance of AVS2 with three different coding configurations AI, RA, and LD, similar to the high-efficiency video coding (HEVC) common test conditions and Bjøntegaard delta bit rate is used for bit rate saving evaluation. The ultrahigh-definition (UHD) and 1080p test sequences are the common test sequences used in AVS, including partial test sequences used in HEVC, such as Traffic (UHD) and Kimono1(1080P), etc. All of these sequences and the surveillance/videoconference sequences used for LD testing are available on the AVS Web site [21].

Table 3 summarizes the rate distortion performance of AVS2 for three test cases. As shown in the table, for RA and AI configurations, AVS2 shows comparable performance as HEVC and outperforms AVS1 with significant bits saving, up to 50% for RA. For surveillance and videoconference video coding, AVS2 outperforms HEVC by 32.1%, and the curves in Figure 11 show the results on two surveillance video sequences. For the coding configurations more reasonable for scene video coding, AVS2's gain is more significant. It should be pointed out that the results are tested with the current AVS2 reference software RD9.2, which is still under optimization, and the performance of AVS2 may be improved further.

CONCLUSIONS

This column gives an overview of the upcoming 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. For surveillance video and videoconferencing applications, AVS2 bridges video compression with machine vision by incorporating smart coding tools, e.g., background picture modeling and location/time information etc., thereby making video coding smarter and more efficient. Compared to the previous AVS1 coding standards, AVS2 achieves significant improvement in coding efficiency and flexibility. AVS2 has been developed in accordance with AVS and IEEE IPR policies to ensure rapid licensing of essential patents at competitive royalty rates. In the development of AVS2, the favorability of licensing terms was also considered in the adoption of proposals for AVS standards, and the formation of a patent pool is expected in the near future.

Several directions are currently being explored for future extensions of AVS2, including three-dimensional video coding and media description for smarter coding. Related standardization work has started in the AVS Working Group.

RESOURCES

AVS documents and reference software can be found in [21]. AVS products information can be found in [22].

ACKNOWLEDGMENT

This research was sponsored by the National Science Foundation of China under award 61322106.

AUTHORS

Siwei Ma (swma@pku.edu.cn) is a professor at the National Engineering Lab of Video Technology, Peking University, China, and a cochair of the AVS Video Subgroup.

Tiejun Huang (tjhuang@pku.edu.cn) is a professor at the National Engineering Lab of Video Technology, Peking University, China, and the secretary-general of the AVS Working Group.

Cliff Reader (cliff@reader.com) is an adjunct professor at the National Engineering Lab of Video Technology, and the chair of the AVS Intellectual Property Rights Subgroup.

Wen Gao (wgao@pku.edu.cn) is a professor at the National Engineering Lab of Video Technology, Peking University, China, and the chair of the AVS Working Group.

REFERENCES

[1] ITU-T, "HSTP-MCTB Media coding toolbox for IPTV: Audio and video codecs," technical paper, ITU-T Study Group 16 Working Party 3 meeting, Geneva, Switzerland, 10 July 2009.

[2] S. Ma, S. Wang, and W. Gao, "Overview of IEEE 1857 video coding standard," in *Proc. IEEE Int. Conf. Image Processing*, Melbourne, Australia, Sept. 2013, pp. 1500–1504.

[3] Q. Yu, S. Ma, Z. He, Y. Ling, Z. Shao, L. Yu, W. Li, X. Wang, Y. He, M. Gao, X. Zheng, J. Zheng, I.-K. Kim, S. Lee, and J. Park, "Suggested video platform for AVS2," in *Proc. 42nd AVS Meeting*, Guilin, China, Sept. 2012, AVS_M2972.

[4] Q. Yu, X. Cao, W. Li, Y. Rong, Y. He, X. Zheng, and J. Zheng, "Short distance intra prediction," in *Proc. 46th AVS Meeting*, Shenyang, China, Sept. 2013, AVS_M3171.

[5] Y. Piao, S. Lee, and C. Kim, "Modified intra mode coding and angle adjustment," in *Proc. 48th AVS Meeting*, Beijing, China, Apr. 2014, AVS_M3304.

[6] Y. Piao, S. Lee, I.-K. Kim, and C. Kim, "Derived mode (DM) for chroma intra prediction," in *Proc. 44th AVS Meeting*, Luoyang, China, Mar. 2013, AVS_M3042.

[7] Y. Lin and L. Yu, "F frame CE: Multi forward hypothesis prediction," in *Proc. 48th AVS Meeting*, Beijing, China, Apr. 2014, AVS_M3326.

[8] Z. Shao and L. Yu, "Multi-hypothesis skip/direct mode in P frame," in *Proc. 47th AVS Meeting*, Shenzhen, China, Dec. 2013, AVS_M3256.

[9] Y. Ling, X. Zhu, L. Yu, J. Chen, S. Lee, Y. Piao, and C. Kim, "Multi-hypothesis mode for AVS2," in *Proc. 47th AVS Meeting*, Shenzhen, China, Dec. 2013, AVS_M3271.

[10] I.-K. Kim, S. Lee, Y. Piao, and C. Kim, "Directional multi-hypothesis prediction (DMH) for AVS2," in *Proc. 45th AVS Meeting*, Taicang, China, June 2013, AVS_M3094.

[11] H. Lv, R. Wang, Z. Wang, S. Dong, X. Xie, S. Ma, and T. Huang, "Sequence level adaptive interpolation filter for motion compensation," in *Proc. 47th AVS Meeting*, Shenzhen, China, Dec. 2013, AVS_M3253.

[12] Z. Wang, H. Lv, X. Li, R. Wang, S. Dong, S. Ma, T. Huang, and W. Gao, "Interpolation improvement for chroma motion compensation," in *Proc. 48th AVS Meeting*, Beijing, China, Apr., 2014, AVS_M3348.

[13] J. Ma, S. Ma, J. An, K. Zhang, and S. Lei, "Progressive motion vector precision," in *Proc. 44th AVS Meeting*, Luoyang, China, Mar. 2013, AVS_M3049.

[14] S. Lee, I.-K. Kim, M.-S. Cheon, N. Shlyakhov, and Y. Piao, "Proposal for AVS2.0 reference software," in *Proc. 42nd AVS Meeting*, Guilin, China, Sept. 2012, AVS_M2973.

[15] W. Li, Y. Yuan, X. Cao, Y. He, X. Zheng, and J. Zhen, "Non-square quad-tree transform," in *Proc. 45th AVS Meeting*, Taicang, China, June 2013, AVS_M3153.

[16] J. Wang, X. Wang, T. Ji, and D. He, "Two-level transform coefficient coding," in *Proc. 43rd AVS Meeting*, Beijing, China, Dec. 2012, AVS_M3035.

[17] X. Wang, J. Wang, T. Ji, and D. He, "Intra prediction mode based context design," in *Proc. 45th AVS Meeting*, Taicang, China, June 2013, AVS_M3103.

[18] J. Chen, S. Lee, C. Kim, C.-M. Fu, Y.-W. Huang, and S. Lei, "Sample adaptive offset for AVS2," in *Proc. 46th AVS Meeting*, Shenyang, China, Sept. 2013, AVS_M3197.

[19] X. Zhang, J. Si, S. Wang, S. Ma, J. Cai, Q. Chen, Y.-W. Huang, and S. Lei, "Adaptive loop filter for AVS2," in *Proc. 48th AVS Meeting*, Beijing, China, Apr. 2014, AVS_M3292.

[20] S. Dong, L. Zhao, P. Xing, and X. Zhang, "Surveillance video coding platform for AVS2," in *Proc. 47th AVS Meeting*, Shenzhen, China, Dec. 2013, AVS_M3221.

[21] AVS Working Group Web Site. [Online]. Available: http://www.avs.org.cn

[22] AVS Industry Alliance Web Site. [Online]. Available: http://www.avsa.org.cn

[SP]