Present and Future Video Coding Standards

Jie Dong and King Ngi Ngan

Department of Electronic Engineering, The Chinese University of Hong Kong
jdong@ee.cuhk.edu.hk,knngan@ee.cuhk.edu.hk

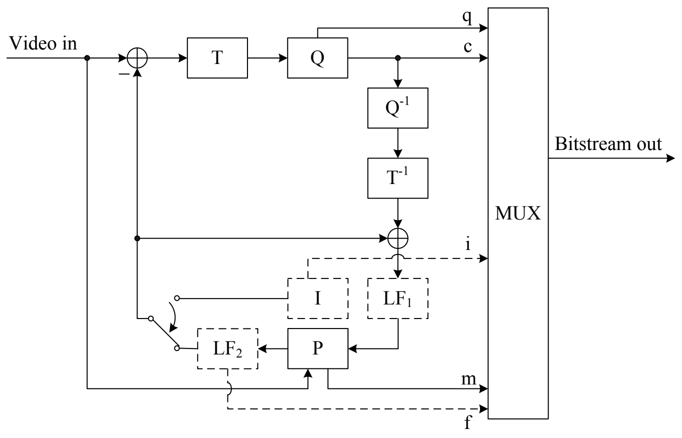
Video coding systems may greatly differ from each other, as they consist of complicated functional modules and each module can be realized by using different techniques. For the interoperability in communication, video coding standards are developed to restrict various video coding systems, such that manufacturers can successfully interwork with each other by producing compliant encoders and decoders, and at the same time still have the freedom to develop competitive and innovative products.

A standard defines a coded representation, syntax, which describes the video in a compressed form. In other words, a standard only specifies the output of the encoder, i.e., the input of the decoder, instead of the codec itself. Although the standard also provides a method of decoding the syntax to reconstruct the video, the manufacturers are free to develop alternative decoders as long as they can decode the syntax and produce the same result as that in the standard.

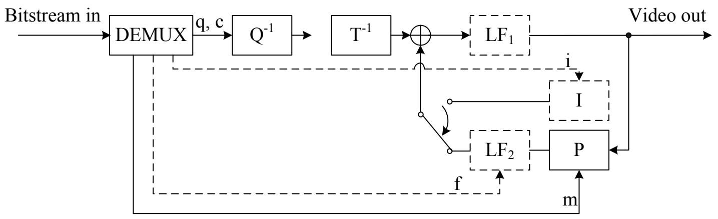
Video coding standards have been developing for about 20 years, driven by applications and advances in hardware capability. This chapter begins with an introduction of the block-based hybrid coding scheme, which is essentially the core of all the video coding standards. In Section 2, the past video coding standards are briefly reviewed, including H.261, MPEG-1, MPEG-2/H.262, H.263, and MPEG-4. The latest standard, H.264/AVC, is the emphasis of this chapter, which is introduced in Section 3, including the new technical developments and profiles favoring a wide range of applications. The recently finalized amendments on scalable video coding (SVC) and multiview video coding (MVC) are another two highlights. Section 4 presents the Audio and Video Coding Standard (AVS) of China, which has received much attention throughout the world, even though it is a national standard. Finally, Section 5 introduces the current topics intensively studied in the standardization groups, which may become the key techniques in the future coding standards.

1 Block-Based Hybrid Coding Scheme

Hybrid coding refers to the combination of motion-compensated prediction (MCP) and transform coding. The former exploits the temporal correlation of



(a)



(b)

T	Transform	q	Quantization parameter
Q	Quantization	c	Quantized transform coefficients
LF_1/LF_2	Loop filter	f	Loop filter on/off
I	Intra prediction	i	Intra prediction mode
P	Picture memory, motion estimation and compensation	m	Motion information

Fig. 1. Block diagrams of the hybrid (a) encoder and (b) decoder

videos, whereas the latter removes the spatial redundancy of the temporal prediction error. The designation "block-based" means each video frame is divided into non-overlapped blocks; each block is the unit where the hybrid coding applies. The phrase "hybrid coding" will refer to block-based hybrid coding scheme here and subsequently in this chapter for convenience.

Fig. 1 shows the generic block diagrams of the hybrid coding scheme for a Macroblock (MB). Note that the modules with dashed boxes are not necessarily included in all standards. Each MB consists of 16×16 luminance pixels and the associated chrominance pixels, depending on the color format, e.g., in 4:2:0 format, Cb and Cr pixels within one MB are both 8×8.

At the encoder (see Fig. 1 (a)), an MB is predicted from a coded frame stored in *P*, known as reference frame. A motion vector (MV) consisting of vertical and horizontal displacements is used to indicate the position of the prediction block, called target block, in the reference frame, and the target block is simply copied for MCP. Besides storing the reference frames, the other function of P is to search the MV, known as motion estimation (ME).

The difference of the current MB and its prediction is input to T, where 2-D transform is applied to compact the prediction error to less coefficients; the coefficients are then quantized in Q. In most standards, the transform is 2-D order-8 Discrete Cosine Transform (DCT) and the quantization is uniform. The quantization stepsize is signaled by the quantization parameter (QP), where a larger QP corresponds to a larger stepsize. The coefficients after quantization are converted to a 1-D array following a zig-zag scan order putting the low frequency coefficients in front of the high frequency ones.

Such hybrid coding process converts the original video to a compressed representation that comprises symbols, called syntax elements (SE). It is followed by a lossless process, known as entropy coding, where all SEs are converted to binary codewords by using variable length coding (VLC) and multiplexed together to a bitstream.

A coded MB belonging to a reference frame should be reconstructed by subsequently applying dequantization and inverse transform and then being compensated by its prediction. A reconstructed reference frame may contain blockiness near the transform block boundaries, which is caused by the quantization and leads to inaccurate MCP. Some standards apply an in-loop deblocking filter on reference frames. The filtering can be enabled before the reconstructed frame is stored in P (see LF_1), e.g., H.264/AVC, or immediately before the target block is used for prediction (see LF_2), e.g., H.261.

At the decoder (see Fig. 1 (b)), the SEs in a bitstream are demultiplexed. The quantized transform coefficients are dequantized, inverse transformed, and compensated by the intra- (optional) or inter-prediction, according to the coding mode. The reference frame used in encoder and decoder should be the same so as to produce identical reconstructed videos.

Using MCP to code an MB is known as inter-mode. It is further classified as P-mode, if the prediction is restricted in the past frame. Otherwise, the inter-mode is known as B-mode, where the prediction is the target block in the past or future frame, or the average of two target blocks from the past and the future frames, respectively. Generally, B-mode can achieve more accurate MCP than P-mode. The coding method without MCP is known as intra-mode, which can be used when inter-mode is not efficient or possible. In early standards, intra-mode directly applies the transform and quantization to original samples. In the recent standards, intra-prediction (see I) is employed in the spatial domain or the transform domain to reduce the spatial redundancy and only the intra-prediction errors are coded. Extending I-, P-, and B-modes to the frame level leads to the concepts of I-, P-, and B-frames.

2 History of Hybrid Coding

The history of hybrid coding is mainly reflected in the evolution of video coding standards, of which the timeline is shown in Fig. 2. The standard series

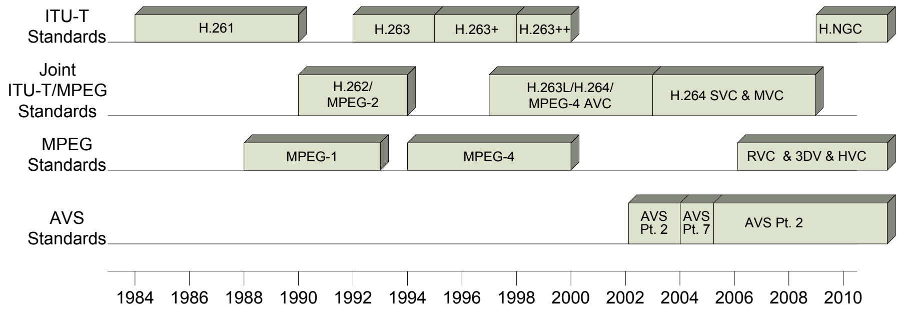


Fig. 2. The timeline of video coding standards

H.26x (x= $1 \cdots 4$) are recommended by the Video Coding Experts Group (VCEG) [1] in ITU-T, and MPEG-x (x= $1, 2, 4$) are developed by the Moving Picture Experts Group (MPEG) [2] in ISO/IEC. A comprehensive survey on the history of MPEG video coding was written by C. Reader [3]. A detailed introduction of the organizations, developments, and techniques of the standards can be found in [4].

2.1 H.261

H.261 [5] was the first standardized hybrid coding scheme published in 1990 by ITU-T, for the interactive video communications over ISDN networks. The supported bit-rate was p$\times$64 kb/s (p= $1 \cdots 30$), depending on the number of ISDN channels used. H.261 was able to operate on QCIF and optionally on CIF, the video formats for visual telephony adopted by ITU-T.

Low complexity was the key factor for the success of H.261, since H.261 was developed at a time when the hardware capability was limited. However, it was soon superseded by H.263, which has higher coding efficiency and greater flexibility, and is still relevant today.

2.2 H.263

H.263 [6] was developed for low bit-rate video communications over analog telephone lines and the Internet and was an evolutionary improvement based on H.261. The initial standard was finished in 1995. Two extensions, nicknamed H.263+ and H.263++, were incorporated into the standard as 21 annexes in 1997 and 2000, respectively.

Baseline H.263 was the basic configuration of the coding algorithm that every compliant H.263 decoder should support. It was distinguished from H.261 by using half-pixel MCP, MV prediction, (level, run, EOB) joint coding for transform coefficients, and so on. Furthermore, H.263 supported more video formats, e.g., sub-QCIF, QCIF, CIF, 4CIF and 16CIF, and a broad range of custom video formats.

Besides baseline H.263, eighteen negotiable coding options (Annexes C-G, I-K, M-V) could be used, either together or separately, subject to certain restrictions. Additional supplemental information (Annexes L and W) might also be included in the bitstream for enhanced display capability and for external usage. A forward error correction method for coded video signal (Annex H) was provided for use, when necessary.

2.3 MPEG-1 Video

MPEG-1 [7] was developed to compress the multimedia content distributed on CD-ROMs at a bit-rate of 1.5 mb/s, catering for the access rate of CD-ROM players. It was conceived to support the "video CD", a format for VHS quality video storage and playback. MPEG-1 typically operated on CIF and SIF videos, and also supported higher rates and resolutions. The functionalities, such as random access, fast forward and rewind, were enabled by the concept of group of pictures (GOP), which starts with an I-frame followed by interleaving P- and B-frames. One can access any GOP without decoding previous GOPs, accomplish fast forward by decoding only I- or I- and P-frames, realize fast rewind by decoding only I-frames in a backward order. Although video CD was not commercially successful, MPEG-1 video had been widely used in many storage and transmission applications.

2.4 MPEG-2 Video

MPEG-2 [8] maintained much similarity to MPEG-1 and also provided the MPEG-1 like functionalities. The main feature distinguishing MPEG-2 from MPEG-1 was the efficient coding of interlaced videos, which was the key enabler of digital storage media and television (TV) broadcast. Two opposite fields in an interlaced frame could be coded in an interleaving manner, known as frame-mode, or separately, known as field-mode. These two modes could be selected once a frame, i.e., picture level adaptive frame/field coding (PAFF), or adapt to the MB level, i.e., MB level adaptive frame/field coding (MBAFF). Coding a field-mode MB basically followed the diagram in Fig. 1, but the reference pictures were fields instead of frames. Other modifications, such as zig-zag scan, were also made.

MPEG-2 mainly operated on the formats specified in ITU-R BT.601 [9] (formerly CCIR 601) with 4:2:0 color format and produced broadcast quality videos at the bit-rates of 4 to 8 mb/s and high quality videos at 10 to 15 mb/s [4]. It also handled HD videos [10] or other color formats at even higher bit-rates. MPEG-2 was a great success with worldwide adoption for digital TV broadcast via cable, satellite and terrestrial channels and for DVD-Video, and finally replaced VHS videotapes.

As a generic coding standard, full MPEG-2 syntax covered many features and parameters to meet a wide spectrum of bit-rates, video formats, quality levels and functionalities. However, certain application required only a subset of the

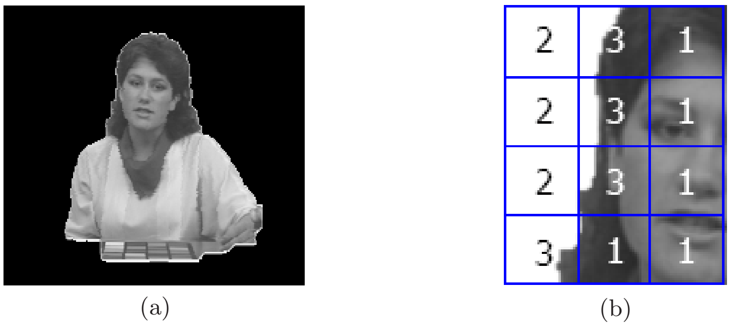


Fig. 3. Content-based coding scheme for arbitrary-shaped VO. (a) The binary alpha map. (b) The opaque, transparent, and boundary MBs

features and consumed limited processing power. To reduce the implementation cost and keep the interoperability, MPEG-2 is the first introducing the concept of profile and level, such that encoders and decoders compliant to the same profile and level could successfully interwork with each other without supporting the full syntax. A profile describes the necessary features; a level indicates the processing capability by specifying the upper limits of spatial resolution, frame rate, and bit-rate. Among the seven profiles in MPEG-2, the Main Profile at the Main Level was the most widely used for TV broadcast.

2.5 MPEG-4 Visual

MPEG-4 was the first attempt to standardize the content-based coding techniques. Although further improving coding efficiency, MPEG-4 Visual [11] went significantly beyond the pure hybrid coding scheme and provided a rich set of tools to code natural and synthetic objects.

The unit of video content is called video object (VO), e.g., a talking person without background. A VO is taken as a sequence of snapshots of an arbitrary-shaped object; its time instances are called video object planes (VOP). As successive VOPs are highly correlated, MCP is also efficient, but the concepts of I-, P-, and B-frame are herein extended to I-, P-, and B-VOP, respectively.

If the shape of a VOP is a frame, i.e., VO is equivalent to a video sequence, the content-based coding scheme is reduced to the conventional hybrid coding. The Simple Profile of MPEG-4 is compatible with the baseline H.263, but MPEG-4 introduces new techniques to improve the coding efficiency, including AC/DC intra-prediction, horizontal scan, global motion compensation (GMC), GMC based on Sprite, 8×8 motion-compensation (MC) block size, and 1/4-pixel MCP.

If VO is of arbitrary shape and location, the coding scheme is extended by additionally coding the shape, represented by a binary or an 8-bit gray-scaled alpha map. The former (see Fig. 3 (a)) defines whether a pixel belongs to an object; the later describes the transparency information. The alpha map is coded on the MB base. For each opaque and transparent MB, labeled 1 and 2 respectively in Fig 3 (b), the encoder just signals whether the MB is inside or outside the VOP. For an MB containing VOP boundaries, labeled 3 in Fig. 3 (b), the alpha map is coded by a context-based arithmetic coder and the VOP texture inside the boundary MB is transformed by the shape-adaptive DCT.

Meanwhile, MPEG-4 took into account specificities of a wide variety of networks and terminals and allows universal access to multimedia information, by employing techniques for a high degree of scalability and error resilience. A comprehensive overview of MPEG-4 Visual can be found in [12], which includes the nature and synthetic video coding, error resilience tools, and scalable coding.

3 H.264/AVC

H.264/AVC [13] is the latest international video coding standard, jointly developed by VCEG and MPEG. In contrast to MPEG-4 Visual covering the ambitious breadth of functionalities, H.264/AVC returns to the narrower and more traditional focus on two goals, efficient compression and robustness to network environments, for generic rectangular-picture camera-shot video content. A comprehensive introduction of H.264/AVC can be found in [14].

This section first illustrates the history of H.264/AVC in Section 3.1. Then, detailed technical designs to improve coding efficiency and facilitate the network transportation are presented in Sections 3.2 and 3.3, respectively. Section 3.4 explains the profiles in H.264/AVC and the corresponding application areas. The recently finalized amendments on scalable video coding (SVC) and multi-view video coding (MVC) are another two highlights, introduced in Sections 3.5 and 3.6, respectively.

3.1 History

The H.264/AVC standardization project was launched by VCEG in early 1998, initially entitled H.26L (the L stood for “long-term”). It was targeted at doubling the coding efficiency in comparison to any other existing video coding standards for a broad variety of applications. The first draft was created in August 1999. In 2001, MPEG issued a Call for Proposals (CfP) with the similar target of H.26L and then adopted the developing H.26L proposed by VCEG as the starting point. Meanwhile, VCEG and MPEG formed a Joint Video Team (JVT) to jointly develop the new standard. The reference software, continuously integrating adopted coding tools, was named joint model (JM). In 2003, the first phase of the standard was finalized and submitted for formal approval, which was later be adopted by ITU-T under the name of H.264 and by ISO/IEC as MPEG-4 Part 10 Advanced Video Coding (AVC) in the MPEG-4 suite of standards. An overview of the first phase of H.264/AVC was presented in [15].

The first version, focusing on entertainment-quality videos with 4:2:0 color format and 8 bit/sample sources, did not support the highest video resolutions used in the most demanding professional environments. To address the needs of professional applications, a continuation of the joint project was launched to add

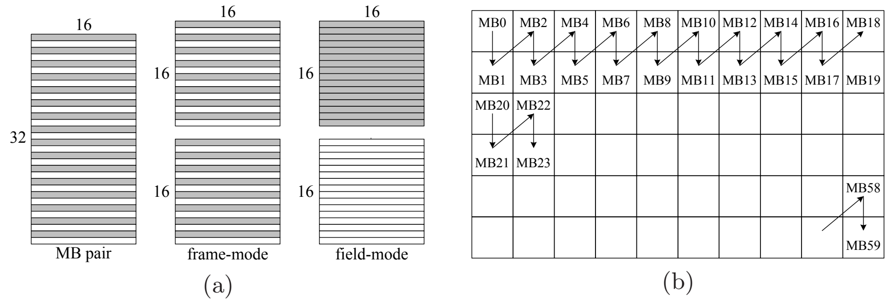

16
32
MB pair
16
16
16
frame-mode
16
16
16
field-mode
(a)
MB0 MB2 MB4 MB6 MB8 MB10 MB12 MB14 MB16 MB18
MB1 MB3 MB5 MB7 MB9 MB11 MB13 MB15 MB17 MB19
MB20 MB22
MB21 MB23
MB58
MB59
(b)

Fig. 4. The concept of MB pair. (a) Coding modes. (b) Coding procedure.

new extensions, named fidelity range extensions (FRExt), to the capabilities of the original standard. FRExt, finalized in 2005, produced a suite of four new profiles, collectively called high profiles. Later in 2007, the breadth of the high profiles were enlarged to eight profiles. In this chapter, the referred H.264/AVC includes the high profiles. An overview of H.264/AVC FRExt is given in [16].

In recent years, SVC and MVC were developed as two amendments to H.264/AVC, driven by the boost of application and network capability. Their histories and backgrounds are introduced in Sections 3.5 and 3.6, respectively.

3.2 Video Coding Layer (VCL)

The VCL design of H.264/AVC essentially follows the hybrid coding scheme, as introduced in Section 1, but significantly outperforms all previous video coding standards. The gain is due to more accurate prediction and more efficient entropy coding, contributed by a plurality of coding elements.

Adaptive Frame/Field Coding

A coded video sequence of H.264/AVC consists of a sequence of coded pictures. In an interlaced video, a coded picture represents either an entire frame or a single field. Two coding modes, PAFF and MBAFF, as in MPEG-2 (see Section 2.4), are also employed. However, as shown in Fig. 4, MBAFF herein makes the frame/field decision at the MB pair level, a 16×32 luminance region, rather than the MB level, so that the basic MB processing structure is kept intact. Fig. 4 (b) illustrates the MBAFF coding process in an interlaced frame.

If an interlaced frame consists of mixed regions, where some regions are moving and others are not, MBAFF coding is efficient to code the non-moving regions in frame-mode and the moving regions in field-mode. However, in the case of rapid global motion and scene change, MBAFF coding is not so efficient as PAFF coding, because one field cannot use the opposite field in the same frame as a reference picture for MCP.

Slice

H.264/AVC supports flexible slice structure, which means the sizes and shapes of slices are not necessarily the multiples of MB rows as in the previous video coding standards. There are five types of slices as explained below, and a coded picture may be composed of different types of slices.

- I-Slice. A slice in which all MBs of the slice are coded as intra-mode.
- P-Slice. In addition to intra-mode, some MBs can be coded using MCP from picture list 0.
- B-Slice. In addition to the coding modes available in a P-slice, some MBs can be coded using bi-directional prediction from picture list 0 and list 1.
- SP-Slice. A so-called switching P-slice that is coded such that efficient switching between different pre-coded pictures becomes possible (see Section 3.3).
- SI-Slice. A so-called switching I-slice that allows an exact match of an MB in an SP-slice for random access and error recovery purposes (see Section 3.3).

Multipicture Reference

H.264/AVC supports multipicture reference, which means more than one previously coded pictures can be used as references for MCP.

Generalized P- and B-Slices

At the decoder, the buffer storing reference pictures is named decoded picture buffer (DPB), which synchronously replicates the buffer of the encoder according to memory management control operations (MMCO), specified in the bitstream. DPB can hold up to 16 pictures, depending on the conformance point and picture size. Any picture type, including B-picture, is allowed to be stored in DPB for future reference, according to the indication in its NAL unit header (see Section 3.3). Thus, the picture type and the referencing capability are decoupled.

Two lists of reference pictures, denoted as list 0 and list 1, can be arbitrarily constructed using the available pictures in the DPB. P-slices use only list 0 pictures for reference, whereas B-slices can use both list 0 and list 1. However, it is not restricted that list 0 and list 1 consist of temporally preceding and succeeding pictures, respectively. Instead, list 0 may contain temporally succeeding pictures and vice versa. It is also possible for list 0 and list 1 to have same pictures. Fig. 5 shows the possible prediction cases for a B-picture, which have never been valid in any previous standards. By this means, the prediction direction and the picture type are decoupled.

Therefore, the only substantial difference between P- and B-slices is that B-slices are coded in a manner that some MC blocks may use a weighted average of two distinct MCP values for building the prediction signal.

Reference Picture Management

The reference pictures in list 0 or list 1 can be classified as short-term and long-term. By default, a short-term picture is a recently coded reference picture, whereas long-term pictures are typically older reference pictures.

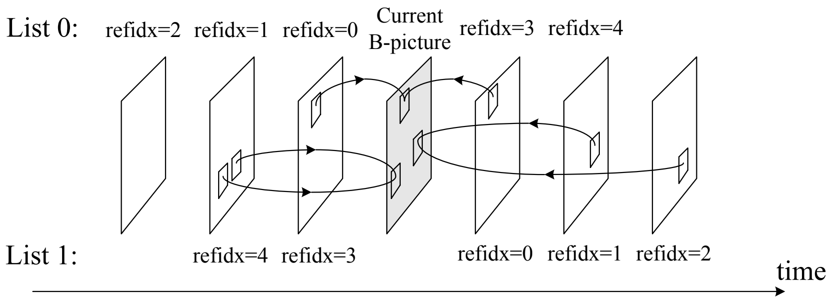


Fig. 5. The construction of list 0 and list 1, and possible bi-directional predictions for B-pictures (reference index is denoted as refidx.)

Each reference picture in a picture list is identified by an index, called reference index, which together with an MV locates a target block for MCP. Among short-term pictures, the one temporally closest to the current picture has reference index equal to 0, and the reference index increases with the temporal distance (see Fig. 5). Each new short-term picture added to the picture list has a reference index equal to 0 and the reference index of each remaining short-term picture is increased by 1. If the number of short-term and long-term pictures reaches the maximum capacity of the picture list, the temporally farthest short-term picture, having the largest reference index, is removed from the DPB.

The operations on long-term pictures, including marking, removing, and replacing, are instructed by MMCO specified in bitstreams. When a short-term picture is marked as a long-term picture, a variable LongTermPicNum is assigned for identification. A long-term picture with smaller LongTermPicNum has a smaller reference index in a picture list and the reference index of long-term pictures starts after all the short-term pictures. Table 1 gives an example of reference picture management for P-pictures, where the size of picture list is five and the current frame number is 100.

The reference picture management introduced in H.264/AVC provides a high degree of flexibility in selecting reference picture, which is constrained only by a total memory capacity bound. This feature is especially efficient for coding videos with periodic transitions, such as interview.

Intra Coding

In contrast to MPEG-1/2/4, where intra-prediction is performed in the transform domain, intra-prediction in H.264/AVC is always conducted in the spatial domain. For the luminance pixels in an MB, three types of partitions are allowed, denoted as INTAR_4×4, INTAR_8×8 and INTAR_16×16; the intra-prediction size for chrominance pixels are fixed to be 8×8.

There are totally nine modes for INTAR_4×4. As shown in Fig. 6, the shaded area is the 4×4 block to predict and the squares labeled with capital letters A-M are the reference pixels in the neighboring coded blocks. Each mode indicates

Table 1. An example of reference picture management for P-pictures

Operation	Reference picture list reference index	0	1	2	3	4
Initial state		-	-	-	-	-
Encode frame 100		100	-	-	-	-
Encode frame 101		101	100	-	-	-
Encode frame 102		102	101	100	-	-
Encode frame 103		103	102	101	100	-
Assign 101 to LongTermPicNum 0		103	102	100	0	-
Encode frame 104		104	103	102	100	0
Assign 103 to LongTermPicNum 4		104	102	100	0	4
Encode frame 105		105	104	102	0	4
Assign 105 to LongTermPicNum 3		104	102	0	3	4
Encode frame 106		106	104	0	3	4

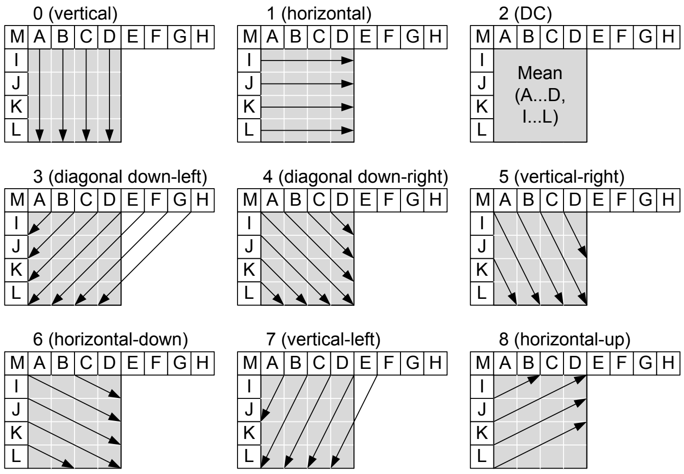


Fig. 6. Nine modes for INTAR_4×4 prediction [14]

a prediction direction except the DC mode, of which the predicted value is the average of all the available reference pixels. INTAR_8×8 for luminance pixels uses basically the same concepts as INTAR_4×4 except the size, but the reference pixels are low pass filtered before used for prediction, in order to reduce prediction error in a larger block.

The intra-prediction with smaller block sizes and more directions is well suited for coding parts of a picture with rich details. On the contrary, for coding very smooth areas, INTAR_16×16 is employed with four modes (see Fig. 7). The 8×8 chrominance pixels within an MB are predicted using a similar prediction as for INTAR_16×16, since chrominance is usually smooth over large areas.

Besides, the standard includes a PCM mode, denoted as I_PCM, with which the values of the samples are sent directly using fixed length coding (FLC).

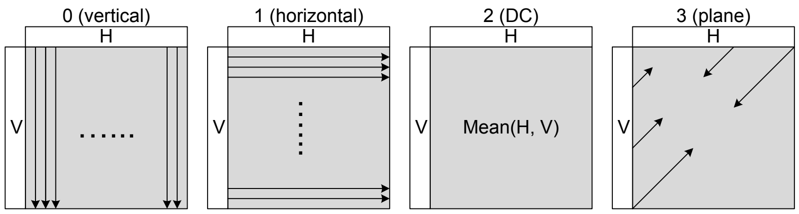


Fig. 7. Four modes for INTAR_16×16 prediction and chrominance intra-prediction [14]

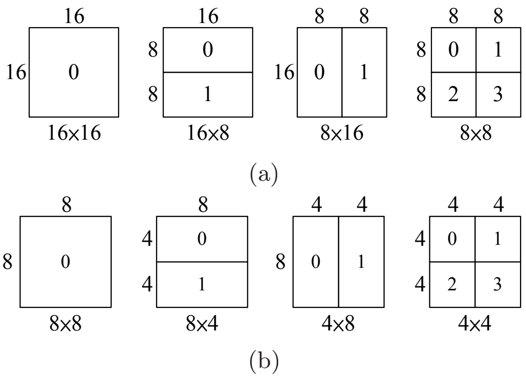


Fig. 8. The partitions of (a) MB and (b) sub-MB for MCP

This mode enables lossless coding for region of interests (ROI) and avoids data expansion, when representing the values in anomalous pictures.

Inter-Coding

Variable MC Block Sizes

In inter-mode, an MB may be partitioned into multiple MC blocks. Fig. 8 (a) shows four partition types for luminance pixels. In case partitions with size 8×8 are chosen, each partition, named sub-MB, can be further partitioned into 8×4, 4×8, or 4×4 MC blocks (see Fig. 8 (b)).

Bigger MC blocks, using less bits to signal the motion information, may result in large MCP errors, and therefore are more appropriate for homogeneous areas of a picture; smaller MC blocks provide more accurate MCP but require more bits for the motion information, so may be beneficial to detailed areas.

MV Prediction

MV in H.264/AVC is predicted from the MVs of the neighboring coded MC blocks and only the prediction error, MVD, is coded. Similar concept has been adopted in previous standards, but the MV prediction in H.264/AVC is improved and more accurate.

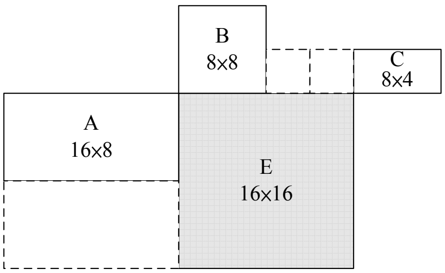


Fig. 9. The adjacent MC blocks involved in MV prediction

As shown in Fig. 9, E is the current block, of which the MV is to predict, and only three adjacent blocks, A, B, and C, are involved to predict the MV of E. The prediction value, MVP, is derived as below.

1. If the size of E is neither 16×8 nor 8×16, the median of the MVs of A, B, and C is used as MVP.
2. For 16×8 MB partition, MVP for the upper 16×8 block is the MV for B and MVP for the lower 16×8 block is the MV for A.
3. For 8×16 MB partition, MVP for the left 8×16 block is the MV for A and MVP for the right 8×16 block is the MV for C.

In case A, B, and C in Fig. 9 are not all available, e.g., outside the current slice or in intra-mode, the derivation of MVP will be modified accordingly.

Interpolation for MCP

For luminance components, the accuracy of MCP is in units of 1/4 distance between pixels and thus the resolution of MV is up to 1/4-pixel. The interpolation for half-pixel samples is separable: a 1-D 6-tap FIR filter [1, -5, 20, 20, -5, 1]/32 is applied for horizontal and vertical directions successively. The samples of 1/4-pixel positions are interpolated using bilinear filter supported by integer pixels and the interpolated half-pixel values. The positions in between chrominance pixels are interpolated by bilinear filtering. Assuming 4:2:0 color format, 1/4-pixel MCP resolution for luminance component corresponds to 1/8-pixel resolution for chrominance components.

Weighted Prediction

In all the previous video coding standards, the target block is directly copied for MCP, or in B-mode, the MCP is the average of the preceding and succeeding target blocks. Such a prediction method is supported by H.264/AVC by default. In addition, H.264/AVC introduces weighted prediction, where the samples of the target blocks are scaled by weighting factors before used for MCP. The weighting factors, w_0 and w_1, can be customized by encoders and transmitted in the slice header, or implicitly derived specially for B-mode in the inverse proportion to the relative temporal distance from the current picture to the reference picture.

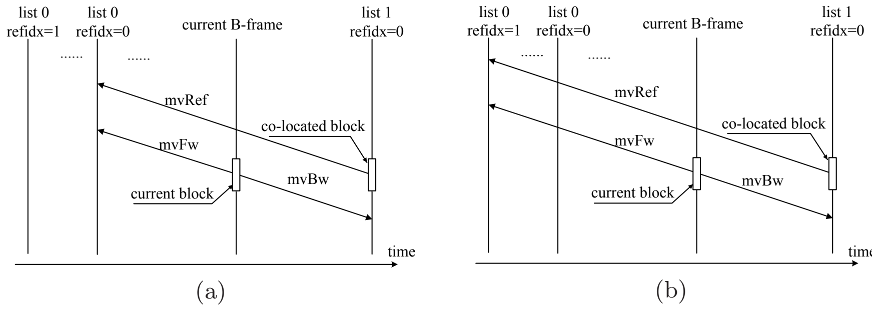


Fig. 10. Derivation of MV and reference index in temporal direct prediction when the MV of the co-located block references the list 0 picture with (a) reference index equal to 0 and (b) reference index equal to 1

Weighted prediction, adaptively controling the relative contributions from target blocks, is especially useful for scene transitions and illumination change.

Improved "Skip" and "Direct" Modes

A skipped MB in previous standards could not move in the scene, which had a detrimental effect for coding video containing global motion. In H.264/AVC, the "Skip" mode for an MB implies no motion in relation to the neighboring MBs rather than absolute rest, such that large areas without change or with constant motion like slow panning can be represented by very few bits.

In P-slices, the "Skip" mode is denoted as P_Skip, with which neither transform coefficients nor motion information are transmitted. The reconstructed MB is exactly the same as its prediction, obtained by using reference index and MVD both equal to 0. In B-slices, the mode similar to P_Skip is called B_Skip, which uses "direct prediction" to infer the MV and reference index. Another mode that also uses direct prediction but requires residual data is known as "Direct" mode, denoted as B_Direct_16×16.

The direct prediction for B_Direct_16×16 and B_Skip modes can be spatial or temporal, as indicated in the slice header. With spatial direct prediction, the MB is predicted as one partition and its MV is derived using the aforementioned MV prediction method. The MV direction can be list 0, list 1, or bi-predictive, depending on the directions of the MVs of the adjacent blocks A, B, and C (see Fig. 9). In some cases, the MV is directly set to zero [13]. In temporal Direct mode, the current MB is partitioned the same way as the co-located MB belonging to the reference picture indexed 0 in list 1. Derivation of the MVs and reference pictures for each partition is illustrated in Fig. 10. The corresponding partitions in the current and co-located MBs use the same list 0 picture. The forward and backward MVs of the current partition are derived by scaling the forward MV of the co-located partition according to the relative temporal distance.

The same spatial and temporal "Direct" concepts can be applied to 8×8 MC blocks, denoted as B_Direct_8×8 mode.

Transform and Quantization

H.264/AVC is the first video coding standard employing an Integer Cosine Ttransform (ICT) rather than DCT, which allows implementation approximations within some specified tolerances [17] (replaced by [18] in 2006). A significant advantage of using ICT is that, with an exact integer inverse transform, the encoder and the decoder perfectly match.

The intra- or inter-prediction error of the luminance pixels in an MB may be split into 4×4 or 8×8 blocks, to which 4×4 and 8×8 ICTs are applied, respectively. The two transforms are adaptively selected for luminance residual data at the MB level, but chrominance pixels are all transformed by the 4×4 ICT. The two transform matrices are given as below.

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \qquad T_8 = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}$$

Given 8-bit input video data, the transforms can be easily implemented using 16-bit arithmetic. Especially, the 4×4 transform is so simple that can be implemented using just a few additions and bit shifts.

As mentioned above, INTAR_16×16 prediction modes are intended for coding of smooth areas and chrominance components are usually smooth over large areas. For that reason, the DC coefficients of the sixteen transformed 4×4 luminance blocks in the MB coded by INTAR_16×16 mode are extracted to form a 4×4 block, which is transformed by a secondary Hadamard transform (HT). Similarly, the DC coefficients of chrominance blocks with all MB types are also transformed using a secondary HT. For 4:2:0 color format, this requires a 2×2 HT. Fig. 11 illustrates such a hierarchical transform procedure for 4:2:0 videos, where the index labeled in each block shows the coding order.

After transform, quantization is applied to meet the target bit-rate. By default, the coefficients in one MB are quantized by the same stepsize determined by QP, an integer ranging from 0 to 51. The value of QP is not linearly related to the stepsize as in all previous standards, but influences the stepsize in such a way that the stepsize increases about 16% for one increament and exactly doubles for every six increments. Rather, the value of QP is linearly related to the actual bit-rate.

Besides the default one, H.264/AVC also supports perceptual-based quantization, which had been a mainstay of prior use in MPEG-2. Although the standard defines a default quantization matrix, an encoder is allowed to specify customized quantization matrices and send them at the sequence or picture level. Such perceptual-based quantization typically does not improve objective

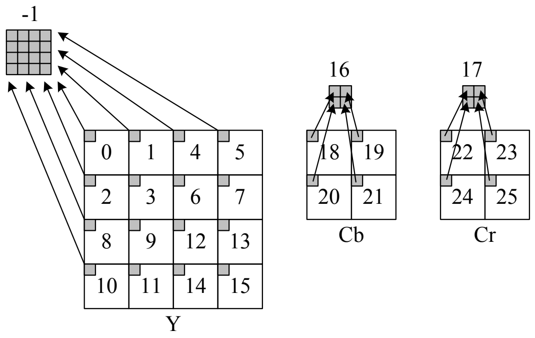


Fig. 11. The hierarchical transform procedure for 4:2:0 videos and the block coding order

quality measured by mean square error (MSE), but it does improve subjective quality, which is eventually the more important criterion.

Entropy Coding

In H.264/AVC, two entropy coding modes are supported. One is a Huffman-like coding, which means each symbol is converted to a binary codeword with integer number of bits, by look-up tables or dynamic codeword construction. The other is the context-adaptive binary arithmetic coding (CABAC). The two modes are switchable at the picture level. The SEs at the higher levels are all coded by FLC or VLC with Exp-Golomb codes. In the mode of Huffman-like coding, the SEs other than the residual data are also coded using Exp-Golomb codes, whereas the residual data are coded using context-adaptive VLC (CAVLC).

Exp-Golomb Codes

Exp-Golomb codes have a generic form of [M zeros][1][INFO] (see Table 2), where INFO is an M-bit field carrying information. Hence, the Exp-Golomb codeword table, which can be constructed in a logical way, is conceptually infinite in length. With Exp-Golomb codes, the standard only defines the mapping from the value of the SE to the look-up index, an unsigned integer denoted as codeNum (see Table 2), according to the probability distribution function (pdf) of the SE.

CAVLC

CAVLC is designed only for coding 4×4 transform blocks. The values and positions of the 1-D 16 coefficients are coded separately, instead of in (level, run) pairs. The coding process is in the reverse zig-zag order, because the trailing non-zero coefficients have high probability to be ±1, whereas the values of leading non-zero coefficients are more random. The events necessary to represent all the information of a block are listed as below. An example of converting the coefficient sequence to events is given in Fig. 12.

Table 2. The Exp-Golomb codes

codeNum	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

Zig-Zag Order		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Coefficients		0	7	0	0	-4	2	0	1	-1	0	0	1	0	1	0	0
CAVLC	levelCode/T1s		12			7	2		0	T1			T1		T1		
	run_before/zerosLeft		1/1			2/3	0/3		1/4	0/4			2/6		1/7		
	←																
CABAC	significant_coeff_flag	0	1	0	0	1	1	0	1	1	0	0	1	0	1		
	last_significant_coeff_flag		0			0	0		0	0			0		1		
	coeff_abs_level_minus1		6			3	1		0	0			0		0		
	coeff_sign_flag		0			1	0		0	1			0		0		
	→																

Fig. 12. Represent a sequence of transform coefficients by symbols with CAVLC and CABAC

1. TotalCoeff. The number of non-zeros coefficients in the block, jointly coded with T1s as one SE coeff_token. In Fig. 12, TotalCoeff equals 7.
2. Tailing ones (T1s). The number of the successive ± 1 coefficients at the end of the coefficient sequence. Note that the maximum value for T1s is 3 and more trailing ± 1–if any–will be coded as regular coefficients.
3. trailing_ones_sign_flag. The sign for each T1.
4. levelCode. Levels of regular coefficients, including the magnitude and the sign. If the level is positive, levelCode equals (level-1)<<1. Otherwise, levelCode equals (-level-1)<<1+1. It is represented by two SEs, level_prefix and level_suffix.
5. total_zeros. The number of zeros before the last non-zero coefficient. In Fig. 12, it equals 7.
6. run_before. Number of successive zeros before each coefficient.

In summary, there are six SEs related to CAVLC, coeff_token, trailing_ones_sign_flag, level_prefix, level_suffix, total_zeros, and run_before. For each SE except trailing_ones_sign_flag, which is a 1-bit flag, multiple VLC tables are designed;

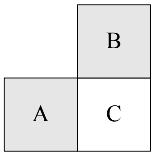


Fig. 13. The context template

each table matches certain conditional statistics. The switching of these tables depends on the local activities, i.e., the context. The meaning of the context for each SE is explained as below.

1. coeff_token. The values of TotalCoeff in neighboring blocks are correlated and can be used as the context. As shown in the context template (see Fig. 13), the table used to code coeff_token in block C is jointly decided by the values of TotalCoeff in A and B.
2. level_prefix and level_suffix. There is no table specified in H.264/AVC for coding levelCode. The codeword for levelCode is the concatenation of the binary representation of level_prefix and level_suffix, both of which have regular forms. The length of the suffix, ranging from 0 to 6, categorizes the codewords into seven conceptual tables, indexed from 0 to 6. The tables in order of ascending indices cater for seven types of contexts from the high frequency coefficients to low frequency coefficients. The switching among the tables is determined by the maximum magnitude of the coded levels in the current block.
3. total_zeros. The choice of the table depends on the TotalCoeff in the current block. Since the summation of total_zeros and TotalCoeff will not exceed 16, large TotalCoeff implies small total_zeros and vice versa.
4. run_before. The choice of the table depends on zerosLeft (see Fig. 12), the number of zeros that have not been processed yet. The range of run_before decreases with zerosLeft.

For coding 8×8 transform blocks, there is no specially designed CAVLC scheme; the 64 coefficients have to be separated into four 4×4 blocks (see Fig. 14), which are successively coded using CAVLC.

CABAC

CABAC is used for coding a broader range of SEs than CAVLC, including slice header, MB header, and residual data. Compared to CAVLC, CABAC typically provides a bit-rate reduction between 5%-15%, although much more computationally complex. The improvement is mainly contributed by three design aspects. Firstly, arithmetic coding allows one to assign a non-integer number of bits per symbol, which is especially beneficial to symbol probabilities greater than 0.5. Secondly, the context modeling allows each SE to have more than one probability models to estimate the conditional probability distributions for different local activities, i.e., different contexts. Thirdly, the probability models can

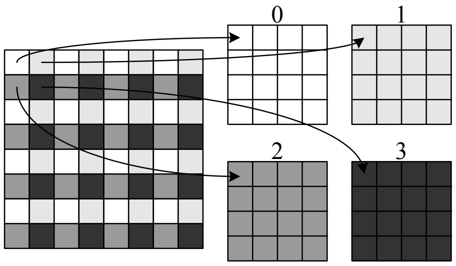


Fig. 14. Separate coefficients of an 8×8 transform block into four 4×4 transform blocks for 8×8 CAVLC

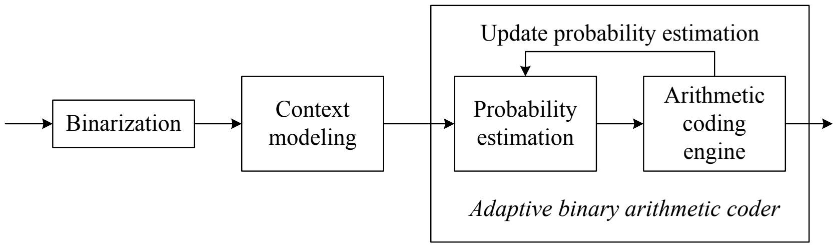


Fig. 15. Generic block diagram of the CABAC entropy coding scheme

be updated to keep track of the actual statistics experienced during the coding process. The diagram of CABAC is depicted in Fig. 15.

The first step is binarization. Since CABAC uses binary arithmetic coding, a non-binary valued SE should first be converted into a binary string. Note that the binary string, still containing much statistical redundancy, is not part of the bitstream. It is only the input to the arithmetic coding engine, of which the output is the compact bitstream.

In the second step of context modeling, each binary decision in a binary string, called “bin”, is assigned an appropriate context model, which estimates the probability distribution of the bin. A context model records two types of information. One is the most probable symbol (MPS), which is either 0 or 1. The other is the probability of the least probable symbol (LPS) P_{LPS}, ranging from 0 to 0.5, is discretized into 64 probability states rather than continuous. In case a context model is used for a single bin or is shared by several bins, there is no need to do the context model selection. Otherwise, the selection among available models is made according to the values of the relevant SEs in the context template (see Fig. 13). After context modeling, each bin as well as its probability distribution, described by the associated context model, are input to the arithmetic coding engine.

The third step is the arithmetic coding. Conventional arithmetic coding is based on the principle of recursive interval subdivision, which introduces multiplications. In H.264/AVC, the arithmetic coding is specified as a multiplication-free low complexity method, using only shifts and table look-ups, which is mainly facilitated by the following three properties.

1. Probability estimation is performed by a transition process between 64 probability states for LPS.
2. The range R representing the current state of the arithmetic coder is quantized to pre-set values. In each iteration of interval subdivision, the new range is obtained by looking up table instead of multiplication.
3. SEs with near-uniform probability distributions are coded by a simplified process, where the context modeling is bypassed,

Each context model has an initial probability estimate defined by the standard and is reset at the beginning of every slice. After the coding of each bin, the context model that has just been used is updated depending on the coded bin is an MPS or LPS. Such an update behavior will change the probability state or even cause the transition between LPS and MPS.

When CABAC is used to code the a transform block, the events describing the values and positions of coefficients are quite different from those of CAVLC. First of all, a flag coded_block_flag is signaled to indicate whether the block contains non-zero coefficients. If so, the coding process is then performed in the order of forward zig-zag scan. As the example in Fig. 12, the value of each coefficient is coded by coeff_abs_level_minus1 and coeff_sign_flag, representing the magnitude and the sign, respectively. The positions of these coefficients are located by a significant coefficient map, which is constructed by significant_coeff_flag and last_significant_coeff_flag. Representing a transform block using such a set of events, which are all binary-valued except coeff_abs_level_minus1, can facilitate the use of binary arithmetic coding.

In-loop Deblocking Filter

In-loop deblocking filters have been adopted in previous video coding standards, such as H.261 and H.263 (Annex J), to reduce the blockiness introduced by discontinuity of motion and block-based transform and quantization. The deblocking filter in H.264/AVC is brought within the MCP loop (see LF_1 in Fig. 1), which leads to not only better subjective and objective qualities but also improved capability to predict other pictures. The deblocking filter is applied to the vertical and horizontal boundaries of 8×8 and 4×4 transform blocks. Each filtering operation affects up to three pixels on either side of the boundary. Fig. 16 shows four pixels on either side of a vertical boundary in adjacent blocks P and Q, where the values of $p0$, $p1$, $p2$ and $q0$, $q1$, $q2$ may be changed by filtering. There are three steps for filtering one block.

First of all, the parameter, boundary strength (BS), is calculated for each boundary, ranging from 0 (no filtering) to 4 (strongest filtering). The value of BS implies the possible extent to which the two adjacent blocks suffer from the blockiness, and thus is used to select an appropriate filter. Fig. 17 shows the BS determination procedure used in progressive video coding, which depends on the

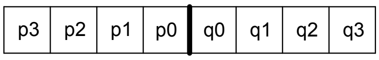


Fig. 16. Pixels on either side of a vertical boundary of adjacent blocks P and Q

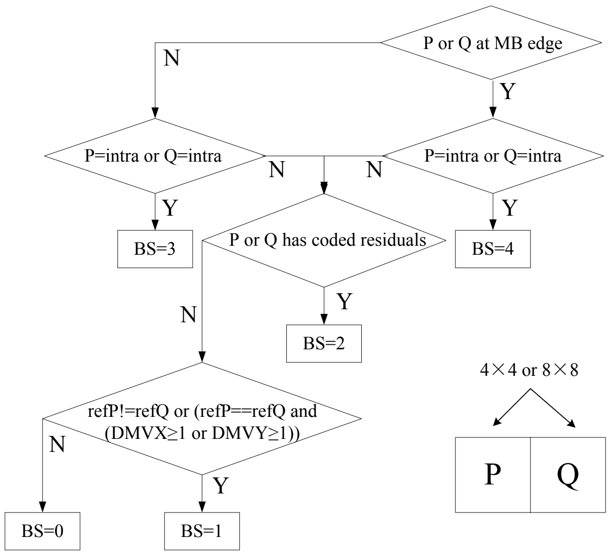


Fig. 17. BS determination procedure used in progressive video coding

boundary location, the coding mode (intra or inter), the existence of non-zero coefficients, and the motion information. In the figure, refP and refQ are the reference information for P and Q, including the direction (list 0, list 1, or bi-prediction) and the reference picture for each direction. "DMVX≥1" means the horizontal distance of the P and Q's target blocks in the same reference picture is greater than or equal to one integer pixel. For coding interlaced videos, the concept is similar, but more logical decisions are involved.

Secondly, while the filter has been selected by BS at the block level, whether to apply such filtering operation is left to be decided for each set of pixels across the boundary (see Fig. 16). The filter decision is made by thresholds $\alpha(x)$ and $\beta(x)$, which decrease with x. The index x is by default the average of the QPs used in P and Q, but can be adjusted by adding an offset transmitted in the slice header. The filtering of $p0$ and $q0$ only takes place if the following condition is true,

$$|p0 - q0| < \alpha(x) \quad \text{and} \quad |p1 - p0| < \beta(x) \quad \text{and} \quad |q1 - q0| < \beta(x)$$

where $\beta(x)$ is considerably smaller than $\alpha(x)$. The filtering of $p1$ and $q1$ takes place if the following condition is true.

$$|p2 - p0| < \beta(x) \quad \text{and} \quad |q2 - q0| < \beta(x)$$

The basic idea is that a relatively large absolute difference between pixels near a block boundary implies blockiness and should therefore be reduced. However, if the magnitude of that difference is so large that it cannot be explained by the coarseness of the quantization, the boundary is more likely to reflect the actual edge in the source picture and should be preserved.

Finally, the filtering operation is applied to where is necessary. The filter varies with BS values, pixel positions, and color components. More details can be found in the standard itself [13].

Designs for 4:4:4 Color Format

As the 4:4:4 color format is usually used in the most demanding video applications, special requirements, such as very high or even lossless fidelity, should be fulfilled. Furthermore, in contrast to 4:2:0 color format, chrominance components are represented in full spatial resolution and thus should be coded using the techniques as powerful as those for luminance components in order to improve the overall performance.

Lossless Coding

The PCM mode in H.264/AVC is developed only to enable the functionality of lossless coding, regardless of coding efficiency. The FRExt includes a transform-bypass lossless mode, where the transform and quantization are bypassed and the intra- or inter-prediction errors are entropy coded in the spatial domain. When a block is coded using intra-prediction with the direction horizontal or vertical, a special method is applied, where the samples along the prediction direction are coded using DPCM. The lossless coding methods introduced herein can also be used for monochrome, 4:2:0, and 4:2:2 videos, but are only enabled in the 4:4:4 profiles (see Section 3.4).

Color Plane Separated Coding

The three color planes in 4:4:4 videos can share the common slice structure and MB coding parameters as for 4:2:0 videos. Alternatively, each color plane can be treated as a separate monochrome video picture and coded individually with its own slice segmentation and mode selections, using the powerful coding tools for luminance components. Furthermore, such a color plane separated coding scheme provides an enhanced parallel processing solution for the encoder by allowing the splitting of the entire encoding processes of the three color planes. The switching of the two coding schemes is indicated by a flag separate_colour_plane_flag in the sequence parameter set.

3.3 Features for Network Transportation

Robustness to data errors and losses and flexibility for operation over a variety of network environments are enabled by a number of new design aspects in H.264/AVC, including the following highlighted features.

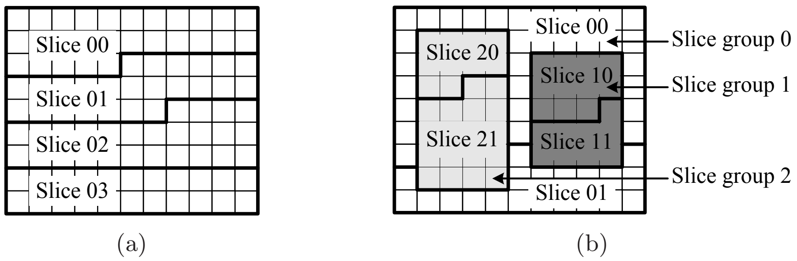


Fig. 18. Subdivision of a QCIF frame into slices (a) without FMO and (b) with FMO.

Parameter Sets

Transmission errors of key information, such as sequence header or picture header, have a severe impact on the decoding process. The parameter set design in H.264/AVC provides robust and efficient conveyance of header information. There are two types of parameter sets, sequence parameter sets (SPS) applied to a coded video sequence and picture parameter sets (PPS) applied to one or more pictures within a coded video sequence. Each VCL NAL unit contains an identifier, referring to the content of the relevant PPS; each PPS contains an identifier, referring to the content of the relevant SPS. In this manner, a small amount of data, i.e., the identifier, can be used to refer to a larger amount of information, i.e., the parameter sets. Separated from the transmission of the video data representation, SPS and PPS can be sent well ahead and repeatedly whenever necessary. It is also possible to convey SPS and PPS “out-of-band” using a more reliable transport mechanism.

Flexible MB Ordering (FMO)

FMO allows a picture to be partitioned into various MB scanning patterns rather than the raster scanning only, by utilizing the concept of slice group. Each slice group is a set of MBs defined by an MB to slice group map, which labels each MB with a particular slice group in the picture. Each slice group can be partitioned into one or more slices, and the MBs in each slice should have the raster scan order inside the belonged slice group. Fig. 18 shows how a QCIF frame is divided into slices. In Fig. 18 (a), where FMO is not used, the whole picture can be considered consisting of a single slice group. In Fig. 18 (b), where FMO is applied, the frame is split into three slice groups, indicated with different shades, and each slice group is further split into two slices. The pattern in Fig. 18 (b) is suitable for coding ROI. Some patterns, such as dispersed and checker-board MB allocations, have been demonstrated to be useful for error concealment in video conferencing applications.

Arbitrary Slice Ordering (ASO)

H.264/AVC enables sending and receiving the slices of a picture in any order relative to each other. This feature, first found in H.263 (Annex K), can improve end-to-end delay in real-time applications, particularly when used in networks having out-of-order delivery behavior, e.g., IP networks.

Redundant Coded Pictures

H.264/AVC allows an encoder to send redundant representations of pictures or regions of pictures, which are typically degraded. Redundant coded pictures have no normative effect on the decoding process and will be decoded only if the relevant primary representation has been lost during data transmission.

Data Partitioning

The concept of data partitioning has been adopted in previous standards, such as H.263 (Annex V) and MPEG-4 Visual. H.264/AVC allows the SEs of each slice to be separated into three different partitions, Partitions A, B, and C, for transmission. Partition A contains the slice header and header data for each MB in the slice; Partitions B and C contain residual data for intra and inter-coded MBs, respectively.

Switching P- (SP-) and Switching I- (SI-) Pictures

In previous video encoding standards, the functionalities of bitstream switching, random access, fast forward/rewind, and error recovery are usually realized by periodically inserting I-pictures. In H.264/AVC, SP- and SI-pictures are developed to provide such functionalities in a more efficient way.

Fig. 19 (a) depicts how to utilize SP-pictures to switch from Bitstream 1 to Bitstream 2. The two bitstreams are assumed to represent the same video sequence at different bit-rates and/or at different temporal resolutions. In each bitstream, primary SP-pictures, such as $S_{1,n}$ and $S_{2,n}$, are periodically inserted to serve as switching points. For each primary SP-picture, the corresponding secondary SP-picture, such as $S_{12,n}$, is generated and sent in response to the switching requirement. A secondary SP-picture has the identical reconstructed values as the relevant primary SP-picture. When switching from Bitstream 1 to Bitstream 2 is required, $S_{12,n}$, predicted from Bitstream 1 but reconstructed as a picture in Bitstream 2, will be sent instead of $S_{1,n}$. Since the reconstructed pictures of $S_{12,n}$ and $S_{2,n}$ are identically the same, the pictures following the switching point, e.g., $P_{2,n+1}$, can be reconstructed without drift, no matter whether the switching has occurred. Note that in fact another secondary SP-picture not shown in the figure, $S_{21,n}$, would be required for switching in the other direction.

Fig. 19 (b) illustrates the switching between bitstreams representing different video sequences by using SI-pictures. SI-pictures are used in the similar way to SP-pictures, except that their relevant secondary pictures are coded in intra-mode. Thus, the switching point enabled by SI-pictures also serves as random access point.

Fig. 19 (c) gives an example of using SP-pictures for error resilience and recovery. When a bitstream is being streamed and there has been a packet loss

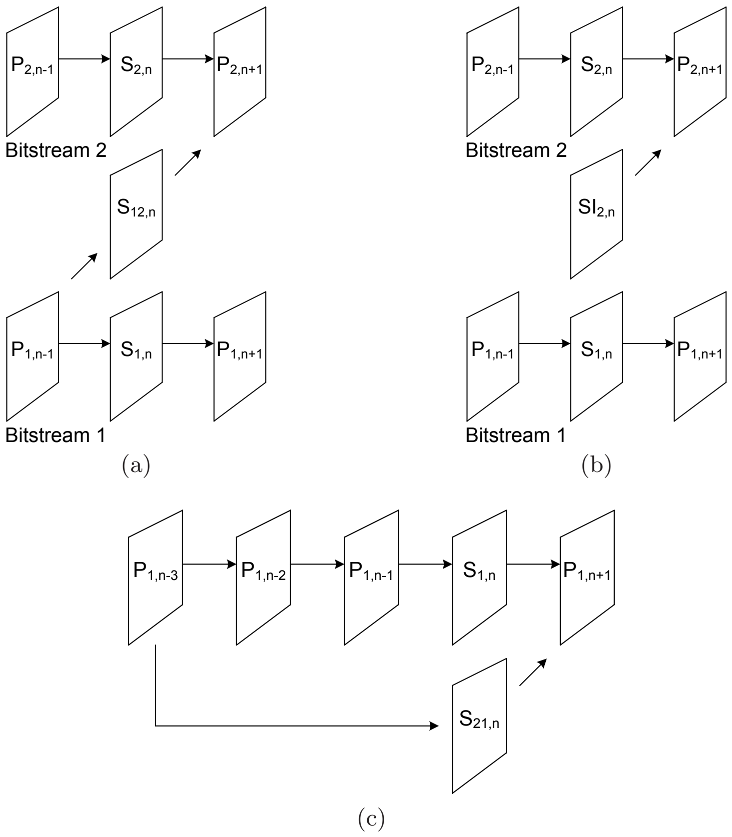


Fig. 19. (a) Switching between bitstreams using SP-frames. (b) Random access using SI-frames. (c) SP-frames in error resilience and recovery.

leading to one or more pictures lost, the client signals the lost pictures to the server, which then responds by sending a secondary SP-picture $S_{21,n}$ of the next primary SP-picture $S_{1,n}$. $S_{21,n}$ references picture $P_{1,n-3}$, which has been correctly decoded. Alternatively, the secondary representation of the next primary SP-picture can be coded as an SI-picture not referencing the previous correctly decoded pictures.

The detailed encoding and decoding processes of SP- and SI-pictures are introduced in [19].

Network Abstraction Layer (NAL)

Rather than forcing a specific bitstream interface to the system as in previous video coding standards, H.264/AVC designs the NAL, which enables simple and effective customization of the method of carrying the video content in a manner appropriate for a broad variety of networks.

Table 3. Types of NAL units

nal_unit_type	Content of NALU	NALU type class
1	Coded slice of a non-IDR picture	VCL
2	Coded slice data partition A	VCL
3	Coded slice data partition B	VCL
4	Coded slice data partition C	VCL
5	Coded slice of an IDR picture	VCL
6	Supplemental enhancement information (SEI)	non-VCL
7	Sequence parameter set (SPS)	non-VCL
8	Picture parameter set (PPS)	non-VCL
9	Access unit delimiter	non-VCL
10	End of sequence	non-VCL
11	End of stream	non-VCL

NAL Unit (NALU)

The coded video data are organized into logical data packets, called NAL units (NALU). Each NALU contains an integer number of bytes, where the first byte is a header and the remaining bytes contain the payload data. The header byte consists of three SEs, forbidden_zero_bit (1 bit), nal_ref_idc (2 bits), and nal_unit_type (5 bits). The nal_unit_type indicates the type of the payload in the NALU. The main NALU types and their corresponding content are shown in Table 3. As can be seen, NALUs are classified into two categories, VCL and non-VCL. The former comprises the data representing the values of the samples in the video pictures; the latter comprises any associated additional information, such as parameter sets and supplemental enhancement information (SEI), which enhances the usability of the decoded video signal but does not affect the normative decoding process.

The nal_ref_idc indicates the reference capability of the payload. If the payload is SPS, PPS, a slice or a slice data partitioning in an IDR or reference picture, which will be referenced for decoding other NALUs, the value of nal_ref_idc is not equal to 0, but no specific value is assigned in the standard. Otherwise, the value of nal_ref_idc equals 0.

NALU Stream

A series of NALUs generated by an encoder is referred to as a NALU stream, which may contain one or more coded video sequences. Each coded video sequence can be decoded independently, given the necessary parameter set information. The NALUs packeting the parameter sets may be conveyed "in-band" or "out-of-band". A coded video sequence consists of a series of access units. The decoding of each access unit results in one decoded picture. At the beginning of a coded video sequence is an instantaneous decoding refresh (IDR) access unit, which represents an I-picture. The presence of an IDR access unit indicates all following coded pictures in decoding order can be decoded without inter-prediction from any picture decoded prior to the IDR picture.

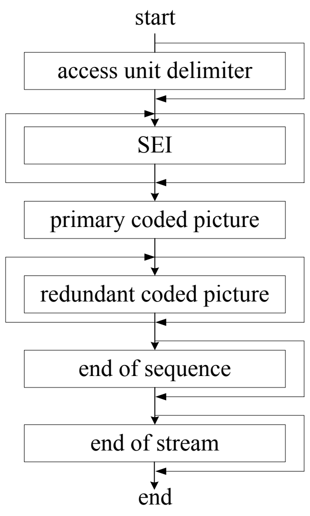


Fig. 20. The structure of an access unit

Fig. 20 shows the format of an access unit. Each access unit contains a primary coded picture, consisting of a set of VCL NALUs. It may also be prefixed with an access unit delimiter to aid in locating the start of the access unit. Some SEI may also precede the primary coded picture. Following the primary coded picture may be some additional VCL NALUs that contain redundant representations of areas of the same video picture, i.e., the aforementioned redundant coded pictures. Finally, if the access unit is the last one of a coded video sequence and/or the entire NALU stream, the NALUs with nal_unit_type equal to 10 and 11 may be presented for indication.

3.4 Profiles and Applications

As introduced in Section 2.4, a profile defines a set of coding tools used in generating a conforming bitstream. Well defined profiles and levels facilitate interoperability between various applications that have similar functional requirements. In the initial version of H.264/AVC finalized in 2003, only three profiles were supported, the Baseline, Main, and Extended Profiles. Fig. 21 shows the coding tools included in these profiles, where the core coding tools cover I-slice, P-slice, intra-prediction, variable MC block sizes, 1/4-pixel MCP, MV prediction, multipicture reference, 4×4 ICT as well as the hierarchical procedure, in-loop deblocking filter, and CAVLC.

The Baseline Profile includes the core coding tools and some error resilience tools. It may be used by conversational services operating typically below 1 mb/s with low latency requirements, such as video telephony, video conferencing, and wireless communications. The Main Profile includes more efficiency-oriented tools to favor entertainment-quality consumer applications operating from 1 to 8 mb/s with moderate latency, such as broadcast, video-on-demand (VOD) via

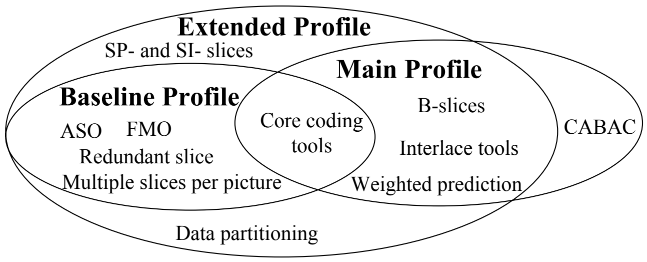


Fig. 21. The coding tools in H.264/AVC profiles

various channels, and high-density storage media. The Extended Profile, a superset of the Baseline Profile, provides improved error resilience and more efficient switching between coded bitstreams. It may be particularly useful for streaming services over the Internet, which typically operates at 50-1500 kb/s and tolerates higher latency.

The union of these three profiles covers all the coding tools in the initial version of H.264/AVC, but excludes those in the FRExt, developed by JVT from 2003 to 2007. The FRExt produces totally eight profiles, collectively named high profiles. Table 4 compares four high profiles, which build upon the design of the Main Profile. As can be seen, the difference in capability among the four high profiles is primarily in terms of the supported bit depths and the chrominance formats. New coding tools further enhancing the coding efficiency are also included, of which some are specially designed for videos with 4:4:4 color format.

The other four high profiles not shown in Table 4 are defined for all-intra coding. Among them, High 10 Intra, High 4:2:2 Intra, and High 4:4:4 Intra Profiles provide the same features and capabilities as High 10, High 4:2:2, and High 4:4:4 Profiles, respectively, except disabling all inter-prediction modes. The last profile, CAVLC 4:4:4 Intra Profile is similar to High 4:4:4 Intra, but the entropy coding is restricted to CAVLC to reduce the complexity.

High profiles were developed to support high quality applications, such as studio camcorders, professional digital video recording and editing systems, digital cinema/large-screen digital imagery, and high fidelity display systems. The four all-intra coding profiles are especially suitable for easy editing of bitstreams. Furthermore, it is anticipated that the High Profile will overtake the Main Profile in the near future as the primary choice for entertainment-quality applications, since the High Profile significantly outperforms the Main Profile without adding much implementation complexity.

3.5 Scalable Video Coding (SVC)

Scalability means parts of the bitstream coded at highest quality and spatial/temporal resolution can be removed in a way that the resulting substream forms another valid bitstream representing the source content with a reduced reconstruction quality of the complete bitstream, e.g., lower fidelity, frame rate, and resolution. This feature is important for bitstreams to adapt to various needs

Table 4. Comparison of the features in four high profiles of H.264/AVC

Features	High	High 10	High 4:2:2	High 4:4:4 Predictive
Main Profile Tools	x	x	x	x
8×8 and 4×4 adaptive transform	x	x	x	x
Perceptual-based quantization	x	x	x	x
Separate Cb and Cr QP control	x	x	x	x
Color plane separated coding				x
Residual color transform				x
Intra residual lossless coding				x
4:2:0 color format	x	x	x	x
4:2:2 color format			x	x
4:4:4 color format				x
8-bit bit depth	x	x	x	x
10-bit bit depth		x	x	x
14-bit bit depth				x

or preferences of end users and to varying terminal capabilities and network conditions. SVC enables this feature by partitioning the data of a bitstream into layers. The base layer is non-scalable and has to be decoded to provide the lowest quality and spatio-temporal resolution, which will be gradually improved with each extra enhancement layer being decoded.

Actually, H.263, MPEG-2, and MPEG-4 have attempted to standardize the technology of SVC. However, the previous designs were not successful in terms of industry adoption, owing to the significant efficiency loss and higher decoding complexity compared with the single-layer coding. In October 2003, MPEG issued a CfP for a new SVC standard, intending to address the shortcoming of the previous designs. The SVC standard was required to enable more flexible transmission in today's highly heterogeneous and time-varying environments and provide comparable rate-distortion (R-D) performance with H.264/AVC single-layer coding. Hence, it is quite natural to build the SVC scheme upon the strong compression foundation of H.264/AVC and employ additional tools to support the required types of scalabilities. In October 2004, the scalability extension of H.264/AVC [20] defeated other proposals and was chosen as the starting point of MPEG's SVC project. In January 2005, MPEG and VCEG agreed to jointly finalize the SVC project as an amendment to H.264/AVC within JVT. The SVC amendment was finally approved in 2007 [21].

The SVC amendment is distinguished from the previous SVC designs by the following three key elements.

- Excellent coding efficiency. For each subset of the scalable bitstream, bit-rate increase at the same fidelity is limited within 10%, compared with the single-layer coding. The detailed R-D performances in a large number of scalability cases can be found in [22].
- Acceptable decoding complexity. A novel approach, known as single MC loop decoding, is adopted, such that the complexity of decoding any subset of the scalable bitstream is comparable with the single-layer decoding.

- Maximal design consistency. The base layer is backward compatible with H.264/AVC, such that the core coding tools of H.264/AVC can be inherited. New tools are added only if necessary for supporting the required types of scalabilities.

Three fundamental types of scalabilities are enabled, i.e., temporal, spatial, and quality scalabilities, which will be discussed respectively in the following sub-sections. An overview of the SVC amendment can be found in [23].

Temporal Scalability

The flexible reference picture management in H.264/AVC (see Section 3.2) enables arbitrary temporal dependencies among successive pictures. Therefore, no change to H.264/AVC is needed to support the temporal scalability with a reasonable number of temporal layers, except the signaling of temporal layers.

Fig. 22 gives three prediction structures, which provide temporal scalability with different temporal dependencies. The picture labeled T_k belongs to the k^{th} temporal layer and T_0 means the base layer. Fig. 22 (a) explains the concept of hierarchical B- or P-pictures, which efficiently provide temporal scalability with dyadic temporal enhancement layers. If the enhancement layer pictures are coded as B-pictures, excellent coding efficiency is achieved. Using P-pictures in enhancement layers can also realize such a temporal coding structure, owing to the decoupling of the picture type and the referencing capability in H.264/AVC. The set of pictures between two successive base layer pictures together with the succeeding base layer picture is referred to as a group of pictures (GOP). Fig. 22 (b) illustrates a non-dyadic hierarchical prediction structure that provides two independently decodable sub-sequences with 1/9-th and 1/3-rd of the full frame rate. Fig. 22 (c) shows a hierarchical prediction structure that does not employ MCP from pictures in the future. This structure provides the same degree of temporal scalability as that in Fig. 22 (a), while its structural delay is reduced to 0, much smaller than 7 pictures for the prediction structure in Fig. 22 (a). However, such a low-delay structure decreases coding efficiency.

Spatial Scalability

The approaches used in previous video coding standards for spatial scalability include intra-layer prediction and inter-layer prediction. With intra-layer prediction, MCP and intra-prediction are employed within each spatial layer as for the single-layer coding. With inter-layer prediction, the reconstructed lower layer signals are upsampled and used as the reference for the MCP of the higher layer. This inter-layer prediction has two disadvantages. Firstly, the statistical dependencies between different layers are not fully exploited, so the R-D efficiency of enhancement layers is relatively low. Secondly, the decoding complexity is high for the enhancement layers, because they cannot be reconstructed until all the depended lower layers are reconstructed.

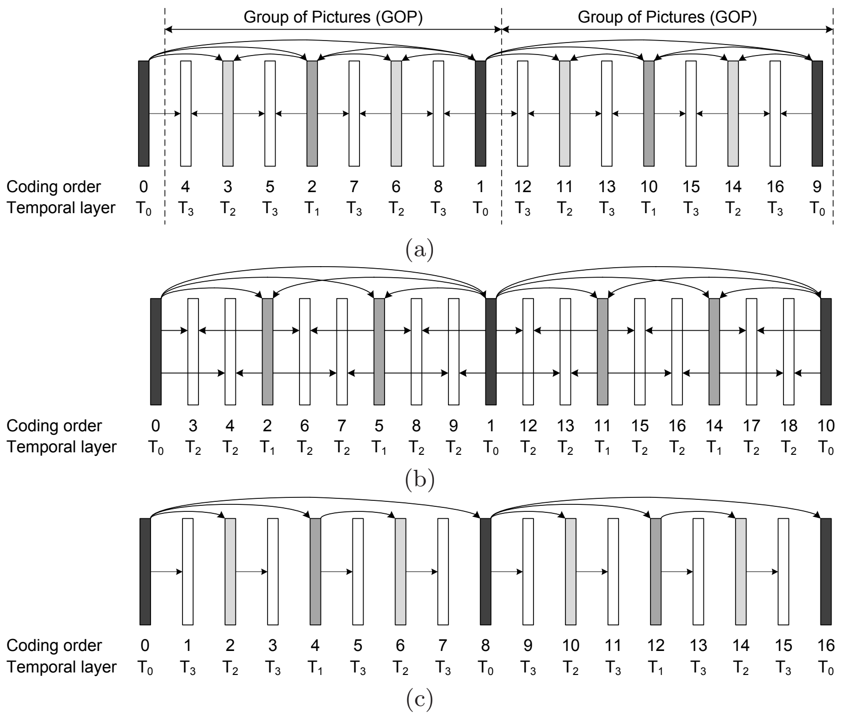

Group of Pictures (GOP)
Group of Pictures (GOP)
Coding order 0 4 3 5 2 7 6 8 1 12 11 13 10 15 14 16 9
Temporal layer T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0
(a)
Coding order 0 3 4 2 6 7 5 8 9 1 12 13 11 15 16 14 17 18 10
Temporal layer T_0 T_2 T_2 T_1 T_2 T_2 T_1 T_2 T_2 T_0 T_2 T_2 T_1 T_2 T_2 T_1 T_2 T_2 T_0
(b)
Coding order 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Temporal layer T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0 T_3 T_2 T_3 T_1 T_3 T_2 T_3 T_0
(c)

Fig. 22. Hierarchical prediction structures for temporal scalability [23]. (a) Hierarchical B prediction structure. (b) Non-dyadic hierarchical prediction structure. (c) Hierarchical prediction structure with a structural encoding/decoding delay of zero.

In the SVC amendment, the conventional intra-layer prediction is followed by using the tools of H.264/AVC and the inter-layer prediction is improved to overcome the aforementioned two disadvantages. In detail, the SVC amendment adopts two new inter-layer prediction modes, inter-layer motion prediction and inter-layer residual prediction, while also keeping the conventional prediction of using the upsampled reconstructed lower layer signals, named inter-layer intra-prediction. Note that the three modes can be chosen at the MB or sub-MB level.

- Inter-layer motion prediction. When the SE base_mode_flag of the MB to be coded equals 1 and the co-located block in the reference layer is inter-coded, the MB is also inter-coded as for the single-layer H.264/AVC coding, but the motion information, including MC block size, MVs, reference index, is completely derived from the co-located block in the reference layer.
- Inter-layer residual prediction. In the transform domain, the coefficients of the co-located block in the reference layer are upsampled and then subtracted from the coefficients of the block to be coded in the enhancement layer. Only the resulting difference, which often has smaller energy than the original residual data, is encoded using entropy coding as specified in H.264/AVC.

- Inter-layer intra-prediction. When the SE base_mode_flag of the MB to be coded is equal to 1 and the co-located block in the reference layer is intra-coded, the co-located block is reconstructed, upsampled, and used as the prediction for the MB to be coded in the enhancement layer. It is further required that the intra-coded MBs in the reference layer should be coded using constrained intra-prediction, of which the reconstruction is independent of any inter-coded MBs.

In the SVC amendment, each spatial enhancement layer can be decoded without completely reconstructing the reference layer. This feature, called single MC loop decoding, significantly reduces the decoding complexity. The inter-layer residual prediction is performed in the transform domain; the inter-layer motion prediction is performed right after the MB header is decoded; only the inter-layer intra-prediction requires reconstructing the co-located block in the reference layer, which, however, typically accounts for a small proportion and does not require reconstructing any neighboring inter-coded blocks because of constraint intra-prediction.

Similar to MPEG-2 and MPEG-4, the SVC amendment supports spatial scalable coding with arbitrary resolution ratios. The only restriction is that neither the horizontal nor the vertical resolution can decrease from one layer to the next. The SVC design further includes the possibility that an enhancement layer picture represents only a selected rectangular area of its corresponding reference layer picture, using a higher or identical spatial resolution.

Quality Scalability

Quality scalability provides the same spatio-temporal resolution as the complete bitstream with a lower fidelity, where the fidelity is often informally referred to as MSE. In the SVC amendment to H.264/AVC, quality scalability can be supported as a special case of the spatial scalability with the resolution ratio between layers equal to 1, which means the aforementioned three inter-layer prediction modes are employed without using the upsampling. In this case, called coarse-grain quality scalable (CGS) coding, the quality refinement of the enhancement layer is achieved by re-quantizing the residual data by a stepsize smaller than that used for the lower layers, and therefore the number of switching rate points is identical to the number of layers.

To increase the flexibility of bitstream adaptation, medium-grain quality scalable (MGS) coding, a variation of the CGS coding, is developed, which modifies the high level signaling to enable the picture (or access unit) level bit-rate adaptation. In detail, in an access unit, formed by the representations with different fidelities for a given time instant, any NALU containing an enhancement layer slice can be discarded. Furthermore, an enhancement layer slice is not necessarily discarded as a whole, since its transform coefficients, allowed to be further distributed to several slices and then packeted by different NALUs, can be discarded partially. Hence, graceful quality degradation is achieved within one enhancement layer slice. In short, MGS significantly increases the granularity

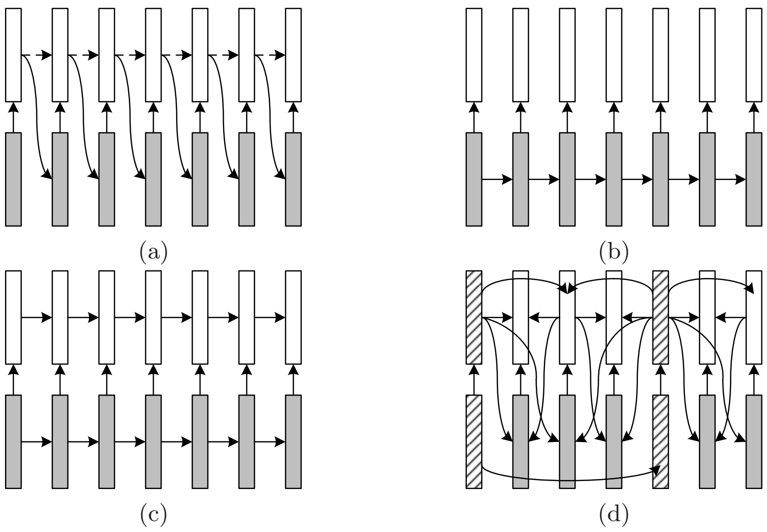
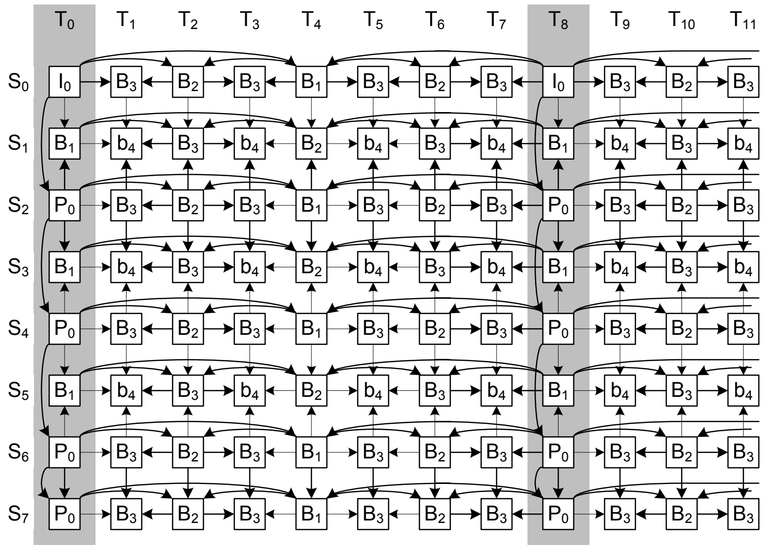


Fig. 23. Concepts for trading off enhancement layer coding efficiency and drift [23]. (a) Enhancement layer control. (b) Base layer control. (c) Two-loop control. (d) Key picture (marked by hatched boxes) concept for hierarchical prediction structures.

of quality scalable coding compared to CGS, although not so flexible as fine-grain quality scalable (FGS) coding, which generates quality scalable bitstreams partially decodable at any bit-rate.

Besides the flexibility of bitstream adaption, the other great concern of quality scalability is the drift, the effect that causes MCP loops at encoder and decoder to lose synchronization. Drift will be inevitably introduced, if the enhancement layer with data loss is used for the MCP of the base layer (see Fig. 23 (a)). Although it is possible to use only the base layer for MCP (see Fig. 23 (b)) such that loss of the enhancement layer does not have any impact on the MCP loop, e.g., the FGS coding in MPEG-4 Visual, the coding efficiency of the enhancement layer is significantly decreased compared to the single-layer coding. The aforementioned CGS coding uses two MC loops as shown in Fig. 23 (c), with which any loss of a quality refinement packet results in a drift for the enhancement layer reconstruction, although the base layer is not influenced. To make a suitable trade-off between drift and enhancement layer coding efficiency, the MGS coding adopts the concept of key picture. Fig. 23 (d) illustrates how the key picture concept is incorporated into hierarchical prediction structures. All pictures of the temporal base layer (marked by hatched boxes) are transmitted as key pictures. In order to limit the decoding overhead, the quality base and enhancement layers of key pictures use exactly the same motion information. The key pictures use the quality base layer as the reference pictures to avoid any drift in the MCP loop of the temporal base layer. In contrast, all pictures in the temporal enhancement layers, i.e., the non-key pictures, typically use the reference with the highest available quality for MCP, which provides a high coding



T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11
S0 I0 B3 B2 B3 B1 B3 B2 B3 I0 B3 B2 B3
S1 B1 b4 B3 b4 B2 b4 B3 b4 B1 b4 B3 b4
S2 P0 B3 B2 B3 B1 B3 B2 B3 P0 B3 B2 B3
S3 B1 b4 B3 b4 B2 b4 B3 b4 B1 b4 B3 b4
S4 P0 B3 B2 B3 B1 B3 B2 B3 P0 B3 B2 B3
S5 B1 b4 B3 b4 B2 b4 B3 b4 B1 b4 B3 b4
S6 P0 B3 B2 B3 B1 B3 B2 B3 P0 B3 B2 B3
S7 P0 B3 B2 B3 B1 B3 B2 B3 P0 B3 B2 B3

Fig. 24. Joint temporal and inter-view prediction structure for the MVC amendment to H.264/AVC [27]

efficiency. Since the key pictures serve as resynchronization points between encoder and decoder reconstruction, drift propagation is efficiently limited within the GOP. The trade-off between enhancement layer coding efficiency and drift can be adjusted by the GOP size and the number of hierarchy stages.

3.6 Multiview Video Coding (MVC)

Multiview video, captured by synchronized cameras from different viewpoints, provides rich 3-D information of a scene and thus expands viewers' experience beyond what is offered by traditional videos. Thanks to the recent advances in acquisition and display technologies, multiview video is foreseen to be the feasible media in consumer domain, including 3DTV and free viewpoint TV (FTV). 3DTV offers a 3-D depth impression of the observed scenery; FTV allows an interactive selection of viewpoint and direction within a certain operating range. The realization of 3-D vision applications will depend on the complete processing chain, including capturing, compression, transmission, display and interactive presentation. Among them, MVC is one of the most challenging technologies, since a tremendous amount of data, in proportion to the number of cameras, needs to be compressed to such an extent that it can be transmitted within the capability of today's network.

MPEG has been investigating the MVC related topics since 2001 in an ad hoc group (AHG), named 3DAV (3-D audio and visual). In July 2005, MPEG issued a formal CfP [24] to address the requirements in [25], based on the evidence brought forward for MVC technology. The proposals in response to the CfP were all backward compatible with H.264/AVC. After one year's evaluation and

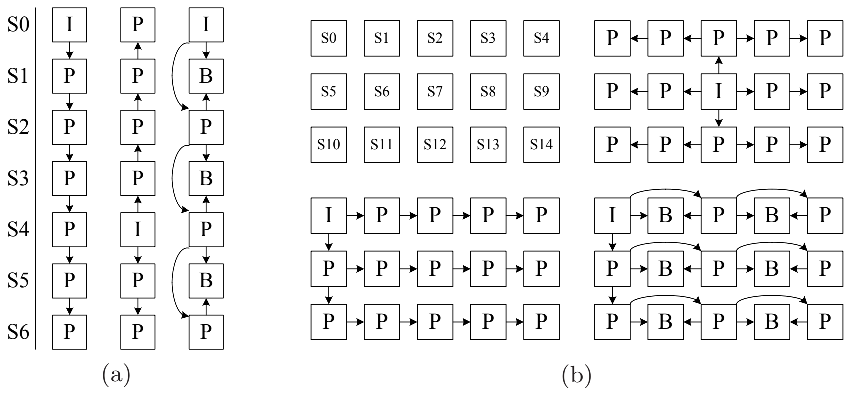


Fig. 25. Inter-view reference dependencies according to camera arrangements [27]

competition, the first MVC model was released. Meanwhile, JVT started to devote part of its effort on the MVC project as another amendment to H.264/AVC. At the time of writing, JVT is at the completion of the MVC amendment [26].

The MVC amendment to H.264/AVC employs inter-view prediction in addition to temporal prediction to remove the inter-view statistical redundancy. Fig. 24 [27] illustrates the joint temporal and inter-view prediction for a multiview video generated by eight linearly arranged cameras. S_n $(n = 0 \cdots 7)$ denotes the individual view sequences and T_n $(n = 0 \cdots 11)$ denotes the consecutive time instances. The first view S_0 employs only temporal prediction as for the single-view coding. More specifically, hierarchical B prediction structure is used for better coding efficiency. As S_0 can be decoded independently of other views, the view scalability is thus provided and S_0 is the base view. For any other view, the temporal prediction structure within the view is the same as that of S_0, whilst inter-view prediction is also enabled, where the pictures from neighboring views at the same time instance are used for MCP. The pictures in the coarsest temporal layer (see the shaded time instances), called key pictures, cannot use temporal prediction, as they provide resynchronization and random access. Therefore, the key pictures in S_0 are all I-pictures and the key pictures in other views can use inter-view prediction. The example in Fig. 24 fully exploits the statistical dependencies, but as a result the video sequences of individual views cannot be processed independently anymore. They have to be either interleaved into one bitstream for sequential processing or signaled and stored in a shared buffer for parallel processing.

Besides the example in Fig. 24, the MVC amendment actually allows a wide variation of joint temporal and inter-view prediction structures, such that the trade-off between the coding efficiency and decoding complexity, including delay and DPB management, can be made. For example, the inter-view prediction can be restricted in key pictures. Furthermore, the inter-view reference dependency can be selected according to the camera arrangement. Fig. 25 (a) shows three possible inter-view reference dependencies for coding the multiview video captured by seven linearly arranged cameras, which result in different complexities. Similarly, Fig. 25 (b) is for a 5×3 camera array.

All the possible joint temporal and inter-view prediction structures are basically the special cases of the SVC amendment. Therefore, no change to SVC is needed except some high level designs, focusing on interface, transport of the MVC bitstreams, and MVC decoder resource management.

4 Audio and Video Coding Standard of China (AVS)

AVS is a suite of standards, including system (Part 1), video (Part 2 and Part 7), audio (Part 3), digital copyright management (DRM) (Part 6), and other supporting standards. The target applications lie in the pivotal fields of Chinese information industry, such as HD digital broadcast, high-density storage media, wireless broad-band multimedia communication, and multimedia streaming. AVS is developed by AVS workgroup, a Chinese standard body established by National Information Industry Ministry in June 2002. This section starts with an introduction of AVS in Section 4.1. Section 4.2 narrows the focus to the technical designs of AVS-Video. The profiles of AVS are presented in Section 4.3.

4.1 Introduction

Video coding standards adopt a large number of patented innovations, which should be licensed at a cost. The standard bodies, e.g., ITU-T and ISO/IEC, deal with such patent issues on a "reasonable and non-discriminatory" (RAND) basis, where the word "reasonable" is very ambiguous. Eventually, the development of necessary patent licensing programs falls to the industry; the manufacturers and end users may risk significantly delayed licensing, unacceptable fees, and complex charging mechanism. For example, the licensing terms for MPEG-4 Visual are so harsh that they have been rejected by most of the market players. In China, the manufacturers produce more than 30 million DVDs and 10 million set-top boxes every year, but make a narrow profit margin because of high license fees. To lower the licensing cost, AVS workgroup was established to develop a suite of national standards that are comprehensive, technically competitive, and affordable for the digital multimedia industry.

The working environment of AVS workgroup is greatly influenced by its Intellectual Property Rights (IPR) policy. An overview of AVS IPR policy is presented in [28]. In short, AVS IPR policy is designed to consider the licensing in parallel with the technical work, which results in two advantages. Firstly, the delay between completing the technical work and the license becoming available is minimized; the licensing should be available soon after the standard is officially approved. Secondly, when deciding which contributions to adopt into the standard, the standard committee does not consider only efficiency and complexity, but also considers the IPR implications, including RAND with royalty-free license, AVS patent pool, and RAND. To avoid high IP cost, some compromises are required, but the benefits of a non-proprietary open standard and the licensing cost savings easily outweigh the small loss in performance. Therefore, AVS IPR policy facilitates early provision of affordable licensing and quick adoption of the standards in markets.

Besides low license fee, AVS also provides competitive performance with succinct design. The coding efficiency of AVS Part 2 is about 2 to 3 times of MPEG-2 and similar to H.264/AVC. In 2006, the Jizhun Profile of AVS Part 2 [29] was officially approved as a Chinese national standard. The standards for audio, system, and DRM have all completed the technical work and are pending for approval.

Nowadays, AVS has been receiving increasing worldwide attention. ITU-T is considering AVS Part 2 as one of the four coding standards for Internet Protocol TV (IPTV). MPEG is also developing the AVS toolbox within the RVC framework (see Section 5.3).

4.2 Highlights of AVS-Video

AVS-Video is essentially similar to H.264/AVC but designed to reduce the implementation cost. Compared to H.264/AVC, some coding tools providing functionalities rather than improving coding efficiency, such as FMO, data partitioning, and SP-/SI- pictures, are not included. To make the design more succinct, the flexibilities of the necessary coding tools, such as intra and inter-prediction, reference picture management, and deblocking filter, are greatly reduced. This section highlights the innovations in AVS-Video, including Part 2 [29]-[31] and Part 7 [32]. A summary of the differences between H.264/AVC and AVS-Video is given in Table 5.

AVS Part 2

Special Reference Pictures

Up to two reference frames are allowed for MCP, which can be used as four fields for field-mode. By default, P-pictures reference the nearest preceding reference pictures, and B-pictures reference the nearest preceding and succeeding reference pictures, both in the display order. To increase the flexibility, three special types of reference pictures are introduced.

- Core picture. Core pictures form the coarsest temporal layer in a bitstream, which has the same concept as the key pictures used for the SVC amendment to H.264/AVC (see Section 3.5).
- Background picture. It is an I-picture, which is expected to be coded at higher quality with noise suppressed for better background prediction. It is stored in the buffer as one of the two reference frames and replaced only upon the reception of the next background frame. This feature is beneficial to video surveillance, where the background is seldom changed, and can be regarded as a special case of long-term picture in H.264/AVC.

Table 5. Comparison of the features in AVS and H.264/AVC

Features	AVS Part 2	AVS Part 7	H.264/AVC
Number of profiles	3	1	11
Color format	4:2:0,4:2:2,4:4:4	4:2:0	4:2:0,4:2:2,4:4:4
Bit depth	8	8	up to 14
Picture type	I,P,B	I,P	I,P,B,SP,SI
Interlace handling	PAFF	N/A	PAFF,MBAFF
Luminance intra prediction	8×8 5 modes	4×4 9 modes	4×4,8×8,16×16 totally 22 modes
No. of reference frame	up to 2	up to 2	up to 16
Reference picture management	background- and core-picture, non-reference P	adaptive sliding window, non-reference P	MMCO-based commands
Prediction in B-picture	forward,backward, Symmetric,Direct	N/A	list0,list1,Direct, bi-predictive
Direct mode	temporal	N/A	spatial, temporal
MV prediction	geometry median	geometry median	median
Interpolation	1/2-pel:[-1,5,5,-1] 1/4-pel:[1,7,7,1]	1/2-pel(Hor.): [-1,4,-12,41,41,-12,4,-1] 1/2-pel(Ver.):[-1,5,5,-1] 1/4-pel: bilinear	1/2-pel: [1,-5,20,20,-5,1] 1/4-pel: bilinear
Minimum MC size	8×8	4×4	4×4
Transform	8×8 PIT	4×4 PIT	4×4/8×8 ICTs
Hierarchical transform	No	No	Yes
Default quantization	64 QPs stepsize doubles for each 8 inc.	64 QPs stepsize doubles for each 8 inc.	52 QPs stepsize doubles for each 6 inc.
Adaptive quantization	perceptual-based	No	perceptual-based
Separate CbCr QP	Yes	No	Yes
Entropy coding	8×8 CA-2D-VLC 8×8 CBAC	4×4 CA-2D-VLC	CAVLC CABAC
Deblocking filter	3 BS levels	2 BS levels	4 BS levels
Lossless coding	No	No	Yes
Weighted prediction	Yes	No	Yes
NAL	No	Yes	Yes
Parameter set	No	Yes	Yes
SEI	No	Yes	Yes
Flexible slice structure	Yes	Yes	Yes
FMO	No	No	Yes
Redundant picture	No	No	Yes
Data partitioning	No	No	Yes

- Non-reference P-picture. P-picture can be marked as a non-reference picture at the picture header, such that temporal scalability is enabled without decoding delay.

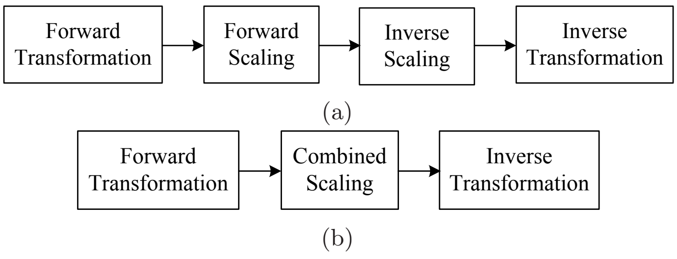


Fig. 26. The flow diagrams of (a) ICT and (b) PIT

"Symmetric" mode in B-pictures

Four prediction modes are used in B-pictures: forward, backward, Direct and Symmetric modes, where the latter two are bi-directional. The Symmetric mode transmits only the forward MV; the backward MV is derived by scaling the forward MV according to the temporal distance. The underlying assumption is that most of the motions in a video sequence are translation and occur within a few neighboring pictures.

Pre-scaled Integer Transform (PIT)

8×8 integer transform is used in order to match the smallest MC block size. The transform matrix given below resembles DCT more than that in H.246/AVC and has higher transform coding gain.

$$T_8 = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}$$

Compared to the ICT scheme in H.264/AVC, of which the flow diagram is shown in Fig. 26 (a), the transform procedure in AVS is further simplified. As shown in Fig. 26 (b), the inverse scaling is moved from the decoder to the encoder and combined with the forward scaling as one single process, named combined scaling. By this means, the inverse scaling is saved and thus no scaling matrix needs to be stored, whilst the complexity of encoders remains unchanged. The simplified transform is named Pre-scaled Integer Transform (PIT), and interested readers are referred to [33] for in-depth expositions.

Adaptive Quantization

AVS Part 2 supports perceptual-based quantization, but the customization of the quantization matrix is restricted by three quantization patterns, as shown in

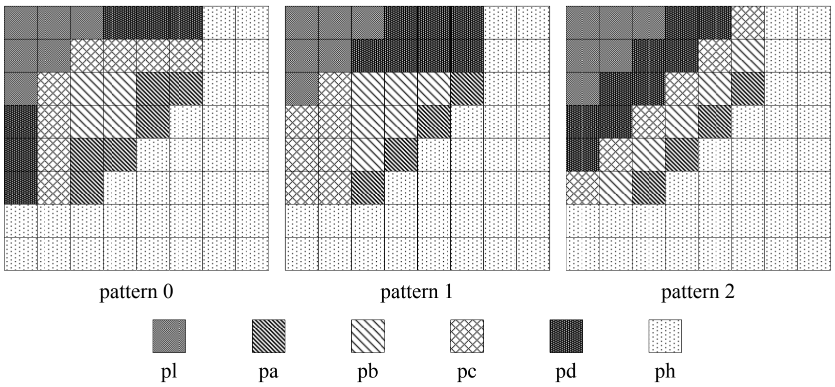


Fig. 27. Three AVS quantization patterns in AVS Part 2

Fig. 27. Each quantization pattern divides a transform block into six specific subbands and the coefficients in one subband use the common quantization weights. There are two default sets of the weights. One set {pl, pa, pb, pc, pd, ph} equal to [135, 143, 143, 160, 160, 213]>>7 is for detail protection; the other set [128, 98, 106, 116, 116, 128]>>7 is for detail reduction. To determine a quantization matrix, three types of information are transmitted in the picture header, the choice of the pattern, the choice of the default set, and the differences between the actual weights and the selected default set if any.

Entropy Coding

Like H.264/AVC, AVS Part 2 supports two entropy coding schemes, Huffman-like coding and arithmetic coding. The former also employs Exp-Golomb codes to code the SEs other than residual data.

In the Huffman-like coding scheme, the residual data are coded using context-adaptive 2-D VLC (CA-2D-VLC), where the levels and runs are jointly coded as (level, run) pairs along the reverse zig-zag scan order and are ended with "EOB". Two main features distinguish CA-2D-VLC from the previous (level, run) coding schemes as in MPEG-2.

Firstly, a plurality of look-up tables is defined to code one transform block. When coding a sequence of (level, run) pairs in one transform block, CA-2D-VLC uses the table indexed 0 for the first pair and switches the tables upon the detection of context changing. The context changing herein means the maximum magnitude of the levels that have been coded in the current block exceeds one of the pre-defined thresholds. There are seven thresholds defined for inter-coded luminance blocks, seven for intra-coded luminance blocks, and five for chrominance blocks, and therefore, totally 19 tables are designed for these three types of blocks.

Secondly, all the codewords in tables are constructed using order-k Exp-Golomb codes, of which the aforementioned Exp-Golomb codes in Section 3.2 are the special cases with k equal to 0 (see Table 6). Exp-Golomb codes with

Table 6. Order-k Exp-Golomb codes

order-k	Codeword	codeNum Range
k=0	1	0
	$01x_0$	1-2
	$001x_1x_0$	3-6
	$0001x_2x_1x_0$	7-14
	...	...
k=1	$1x_0$	0-1
	$01x_1x_0$	2-5
	$001x_2x_1x_0$	6-13
	$0001x_3x_2x_1x_0$	14-29
	...	...
k=2	$1x_1x_0$	0-3
	$01x_2x_1x_0$	4-11
	$001x_3x_2x_1x_0$	12-27
	...	...
k=3	$1x_2x_1x_0$	0-7
	$01x_3x_2x_1x_0$	8-23
	$001x_4x_3x_2x_1x_0$	24-55
	...	...

larger orders favor flatter-shaped pdf and vice versa, whereas the mapping from a (level, run) pair to a codeNum only reflects a relative probability.

The arithmetic coding is named context-based arithmetic coding (CBAC), which basically has the same diagram as CABAC in H.264/AVC (see Fig. 15). The main difference between CBAC and CABAC lies in two aspects. Firstly, the arithmetic coding engine performs the interval subdivisions, i.e., probability update, in the logarithmic domain, where multiplication in the physical domain is equivalent to addition. Therefore, the probability update of the arithmetic coding is approximated by additions and shifts only. Secondly, the transform block is coded as a sequence of (level, run) pairs as in CA-2D-VLC and the context selection is jointly determined by the maximum magnitude of the levels that have been coded in the current block and the position of the current level in the zig-zag order.

AVS Part 7

Flexible Reference Frame

H.264/AVC can handle up to 16 reference frames by complicated MMCO-based commands, and the reference pictures are categorized into short-term and long-term ones. In AVS, only two reference frames are used, so the operation to realize such a categorization is significantly simplified. The concept of adaptive sliding window is introduced, of which the window size can be one or two, selected at the frame level. If the window size equals two, both reference frames will be involved in the first-in-first-out sliding window operation; the temporally farther

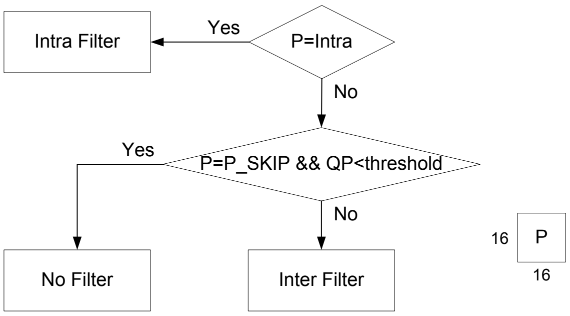


Fig. 28. The deblocking filter mode decision in AVS Part 7

reference frame will be pushed out of the DPB. Otherwise, only the temporally nearer frame is in the sliding window, which will be removed from the DPB by the just arrived reference frame. Obviously, window size equal to one makes the reference frame outside the sliding window act as a long-term frame.

All the high-level designs and error resilience tools in AVS Part 7, including this reference frame management, are discussed in depth in [34].

Transform Matrix

The transform in AVS Part 7 is also PIT, and the transform matrix is given as below.

$$T_4 = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 3 & 1 & -1 & -3 \\ 2 & -2 & -2 & 2 \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

Simplified Deblocking Filter

The deblocking filter in AVS Part 7 is significantly simpler than that in Part 2 or H.264/AVC. Only two filtering modes, decided at the MB level, are used, whereas H.264/AVC uses four modes (BS) decided at the block level. Fig. 28 shows the procedure of determining filtering mode, which involves much less coding information and logical decisions than Fig. 17. The "threshold" in the figure is equal to 40 by default and can be modified by transmitting an offset in the frame header. Furthermore, each filtering operation affects only two pixels on either side of the boundary, and thus the pixels filtered for neighboring boundaries do not overlap. This enables parallel operations during the deblocking process.

Interested readers are referred to [35] for the details of all the low complexity coding tools in AVS.

4.3 Profile and Application

AVS Part 2 contains three profiles, Jizhun (Base) [29], Jiaqiang (Enhanced) [30], and Shenzhan (Extended) [31] Profiles. Fig. 29 shows the coding tools included

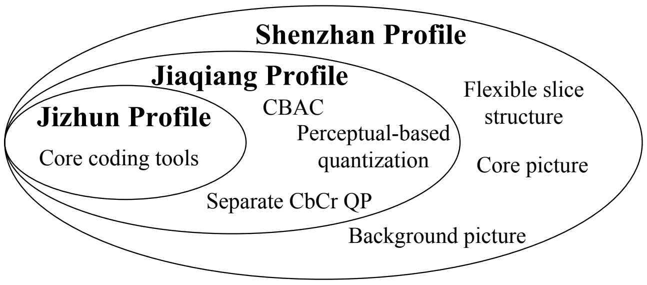


Fig. 29. The coding tools in AVS profiles

in different profiles, where the core coding tools are I-, P-, and B-pictures, 8×8 intra-prediction, variable MC block sizes, 1/4-pixel MCP, MV prediction, two reference frames, 8×8 PIT, deblocking filter, CA-2D-VLC, weighted prediction and PAFF.

Jizhun Profile mainly focuses on digital TV applications, such as terrestrial TV, IPTV, satellite TV, as well as storage media. As the enhancement of Jizhun Profile, Jiaqiang Profile adds CBAC and adaptive quantization, in order to address the requirements of high quality video applications, such as digital cinema. Shenzhan Profile is developed specially for video surveillance applications. Features of flexible slice structure, core picture, and background picture are incorporated, which improve the capability of error resilience, temporal scalability, and friendliness to the surveillance environment.

AVS Part 7 has been developed to meet the needs of mobile video applications, such as interactive storage media, video services on wireless broad-band and packet networks. AVS Part 7 has only one profile, named Jiben (Basic) Profile, which contains all the features in this part.

5 Future Video Coding Standards

At the time of writing, the two standard bodies, MPEG and VCEG, are moving forward to develop the future video coding standards based on the foreseen requirements. This section gives a brief introduction of the requirements and the ongoing standardization efforts.

5.1 High-Performance Video Coding (HVC)

Thanks to the technology evolution, more and more video materials with increased quality and spatio-temporal resolution (see Table 7) will be captured and distributed in the near future [36]. It is anticipated that the bit-rate produced by the current coding technology will go up faster than the increased capacity of the wireless or wired network infrastructure. Therefore, a new generation of video compression technology aiming at sufficiently higher compression capability rather than rich functionalities is required. Both VCEG and MPEG have been studying the feasibility of such a new standard.

Table 7. The video formats to be supported by high-performance video coding

Frame rate	typically 24-60 fps, up to 172 fps
Spatial resolution	VGA/720p/1080p/4K×2K
Color space	YCbCr or RGB
Color format	4:2:0/4:2:2/4:4:4
Bit depth	typically 8 bits, up to 14 bits

H.NGVC in VCEG

The project VCEG is currently working on is named H.NGVC, standing for "Next Generation Video Coding". H.NGVC could mean either an extension of H.264/AVC or a new standard, depending on the technical design. In January 2009, VCEG drafted a requirement for H.NGVC, which was basically consistent with the specification in Table 7. Regarding the coding efficiency, H.NGVC is expected to provide 50% bit-rate reduction at the same subjective quality compared to H.264/AVC.

Actually, VCEG has been seeking evidence regarding the possibility of a major objective performance gain over H.264/AVC ever since 2005. To better evaluate the techniques and retain the progress, key technical area (KTA) is developed as the reference software, which uses JM11 as the baseline and continuously integrates promising coding tools. Here listed are the techniques having been adopted in KTA so far.

- High-resolution MCP. The resolution of MV is increased from commonly used 1/4-pixel to 1/8-pixel, which has been proved especially efficient for low resolution video sequences.
- Adaptive interpolation filter (AIF). The coefficients of AIF are customized at the picture level and coded as the side information. Different AIF techniques make different approximations to the optimal filter that minimizes the prediction error energy, such as reducing the support region, imposing the symmetry constraints, separating the 2-D operation, and quantizing the filter coefficients, so as to reduce the side information and complexity.
- Adaptive quantization matrix selection (AQMS). The quantization matrix is designed on-the-fly or selected from a pre-defined candidate pool at the MB level. The selection is based on the criterion of R-D cost and signaled in the bitstream.
- Adaptive prediction error coding (APEC). With the increased prediction accuracy, the correlation of the residual signals decreases, so sometimes the transform becomes inefficient for energy compaction. APEC allows the residual data to be coded in either transform or spatial domain, decided and signaled down to the transform block level. In spatial coding mode, prediction errors are coded by specially developed quantization and entropy coding, without transform.
- Competition-based MV prediction. Instead of having one single MV predictor as in H.264/AVC, a set of spatial, temporal, and spatio-temporal predictors compete with each other; the predictor resulting in the lowest R-D cost wins.

Such a competition-based scheme is also employed to derive the MV for Skip mode. No matter what the MB type is, the selected MV predictor is signaled in the bitstream.

- Mode-dependent directional transform (MDDT). For intra-prediction modes with strong directionality, e.g., vertical mode and horizontal mode, corresponding MDDTs are derived from KLT to favor the high energy along the directions. The type of MDDT is coupled with the selected intra-prediction mode, so is not explicitly signaled. However, the memory to store all predefined MDDT bases is up to 1.5Kb.
- Extended block sizes for MCP and transform. The MB size is enlarged to 32×32 or 64×64, and the sizes for MCP are scaled accordingly. A 2D order-16 transform are also adopted for the residual blocks produced by MC block larger than or equal to 16×16. The transform matrix is obtained by scaling the transform matrix of 2D order-16 DCT by the factor 128 and rounding. This technique is developed to serve the upcoming applications of ultra-high definition TV (UDTV).

Meanwhile, VCEG keeps seeking the best combination of these coding tools, as their performance gains are not necessarily addictive.

HVC in MPEG

In October 2008, MPEG issued a Call for Evidence [37], which means a more rigorous evaluation phase for potential technologies. The target technology is not necessarily built on the top of the state-of-the-art standard. An absolutely new paradigm is also possible, as long as sufficient performance gain is observed. Both subjective and objective evaluation methodologies will be employed. Positive evaluation result may lead to a formal CfP and the new standard development with a tentative name high-performance video coding (HVC).

5.2 3D Video Coding in MPEG

The MVC amendment to H.264/AVC introduced in Section 3.6 efficiently reduces the bit-rate compared to multiview simulcast, but the bit-rate proportionally increases with the number of views, because the required views at the receiver all need to be coded. Therefore, the supported maximum number of views is limited due to the constraints of transmission and processing capabilities.

MPEG envisions that 3D media applications, providing the viewers more immersed experience, will become reality in the next few years [38]. The viewers will perceive stereoscopic videos at arbitrary viewpoint by using auto-stereoscopic displays, head-mounter displays (HMD) or head tracking. These applications require producing a large number of views for display, which go beyond the capabilities of the MVC amendment to H.264/AVC. In April 2007, MPEG launched an exploration on a new multiview video presentation format, named 3-D video (3DV), which bears explicit 3-D information, i.e., the depth, and a few channels

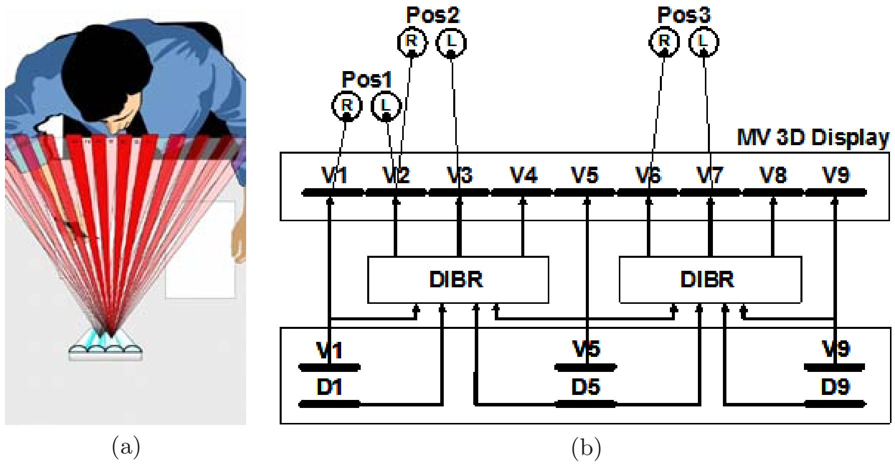


Fig. 30. Illustration of the 3DV application scenarios [39]. (a) An auto-stereoscopic display. (b) Simultaneously presenting nine views by decoding and rendering.

of videos, and supports synthesizing high-quality views for continuous viewpoints. By this means, the transmission rate, which is almost constant, is decoupled from the number of output views.

Fig. 30 illustrates the 3DV application scenarios [39]. Fig. 30 (a) shows an auto-stereoscopic display presenting nine views simultaneously to the users, where the neighboring views, forming a stereo pair, provide 3-D perception while the user is moving within a narrow angle. As shown in Fig. 30 (b), only three views (V1, V5, V9) with the associated depth data (D1, D5, D9) are received, and the other six views to display are generated by depth image based rendering (DIBR).

In the whole architecture of 3DV communication, three designs are considered by MPEG for standardization, i.e., 3DV data format, decoder, and view interpolation.

- 3DV data format, including multiview video, camera parameters, depth data, and additional information (if necessary to make the system more efficient), is the output of the decoder and the input for view interpolation. 3DV data format should be hardware-independent to ensure wide applicability and interoperability.
- A decoder reconstructs 3DV data format in a normative manner.
- The interpolation module interpolates the views for display, according to the standardized algorithm, the decoded 3DV data, and some parameters from the display module, such as size of image, number of views, viewpoints, and frame rate. However, at the moment, the standardization of view interpolation has been receiving arguments concerning the evolution of rendering technology and unnecessary cost for certain cases.

At the time of writing, the formal standardization activity has not been launched; MPEG is exploring the related techniques by four exploration experiments (EE), focusing on depth estimation/generation, view synthesis, 3DV data format definition, and coding experiments, respectively. Especially, the fourth EE aims at getting an impression of how the depth map coding affects the quality of synthesized views. Compared to the MVC amendment to H.264/AVC, the preliminary results show that utilizing depth information improves the synthesis quality to some extent but rather sequence-dependent. Depth rate ranges from 10% to 50% of the texture rate.

5.3 Reconfigurable Video Coding in MPEG

Having successfully developed many video coding standards, MPEG initiated another standardization activity, named reconfigurable video coding (RVC), motivated by the following considerations [41].

Firstly, nowadays a multimedia device quite often should support not only the latest standard but also several legacy ones, as well as their multiple profiles. The standards are realized case by case without recognition or exploitation of the commonality among their coding tools. Such implementation redundancy can be efficiently reduced by modularizing and reusing the coding tools in a multi-codec device.

Secondly, the traditional standardization procedure, typically taking two to three years, is too slow to satisfy the rapidly changing landscape and requirements of media coding applications. Hence, standardizing new technologies at the coding tool level instead of the codec level offers a faster path to improving MPEG standards and better addressing the growing requirements.

Thirdly, coding tools that have to be implemented by a compliant decoder may not be all used in decoding the bitstream produced by a certain encoder, as encoder can choose the coding tools more freely. The decoder will become more customized, if dedicatedly configured based on the corresponding encoder.

RVC, formally launched by MPEG in January 2006 [40], is actually a high level specification model for direct and efficient codec synthesis. In other words, RVC offers a framework capable of constructing a video codec by configuring its coding tools, thus enabling a dynamic development, implementation and adoption of standardized video coding solutions. The RVC framework, comprising video tool library (VTL) and decoder description (DD) [41], is introduced as below.

A normative VTL consists of function units (FU) drawn from the existing MPEG coding standards, so-called MPEG toolbox. VTL is specified by a textual specification and the corresponding reference software. The reference software is provided using the specification language RVC-CAL, standardized by MPEG, where the data flow components are called actors. An actor encapsulates its own state and thus cannot read or modify the state of any other ones. The only interaction between actors is via messages, known as tokens in RVC-CAL, which are passed from an output of one actor to an input of another. The behavior of an actor is defined in terms of a set of actions, such as reading input tokens,

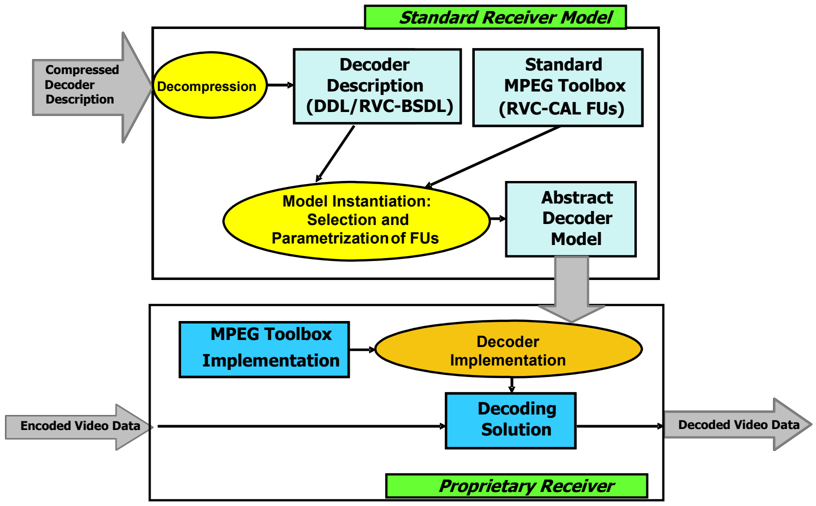

Standard Receiver Model
Compressed Decoder Description
Decompression
Decoder Description (DDL/RVC-BSDL)
Standard MPEG Toolbox (RVC-CAL FUs)
Model Instantiation: Selection and Parametrization of FUs
Abstract Decoder Model
MPEG Toolbox Implementation
Decoder Implementation
Decoding Solution
Encoded Video Data
Decoded Video Data
Proprietary Receiver

Fig. 31. The conceptual process of deriving an abstract decoder model in the MPEG RVC specification [41]

modifying internal state, producing output tokens, and interacting with the underlying platform on which the actor is running. At most one action is active at any time instance.

A decoder within the RVC framework should implement at least one VTL. At the moment, the defined VTLs include MPEG-2 Simple Profile and Main Profile, MPEG-4 Visual Simple Profile, MPEG-4 AVC Baseline Profile and SVC Baseline Profile. MPEG continues working on defining the remaining coding tools in MPEG toolbox for RVC-oriented FUs. Non-MPEG VTLs, such as AVS coding tools, are also supported by RVC framework, but MPEG will not take the responsibility of specification, conformance, or any technical qualification assessment.

DD is coded and transmitted together with the encoded video data, which carries two types of information. The first type describes how FUs are connected and parameterized. The second type, which describes the bitstream syntax structure by using the profiled Bitstream Syntax Description Language (BSDL), called RVC-BSDL, is used to synthesize a bitstream parser.

With the introduced normative elements, an abstract decoder model (ADM) is constituted (see Fig. 31). ADM, a behavioral model of decoder configuration, is platform-independent, whereas the decoder implementation is platform-dependent, as it is free to substitute proprietary VTLs having standard I/O interface for standard VTLs. Finally, an instantiation of decoding solution is established, which is actually a dedicated decoder to reconstruct the received video bitstream.

References

1. Terms of reference, ITU-T SG16/6 web site, `http://www.itu.int/ITU-T/studygroups/com16`
2. Terms of reference, MPEG home paper, `http://www.chiariglione.org/mpeg/`
3. Reader, C.: History of MPEG video compression. JVT document JVT-E066 (October 2002)
4. Wang, Y., Ostermann, J., Zhang, Y.-Q.: Video Processing and Communications. Prentice-Hall, New Jersey (2002)
5. ITU-T Recommendation H.261, Video codec for audiovisual services at p×64 kbits (1993)
6. ITU-T Recommendation H.263, Video coding for low bit rate communication (1998)
7. ISO/IEC IS 11172-2, Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2: Video (1993)
8. ISO/IEC IS 13818-2, Information technology – Generic coding of moving pictures and associated audio information: Video (1996)
9. ITU-R Recommendation BT.601-5, Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios (1995)
10. ITU-R Recommendation BT.709-5, Basic parameter values for the HDTV standard, for the studio and for international program exchange (2002)
11. ISO/IEC IS 14496-2, Information technology – Coding of audio-visual objects – Part 2: Visual (1998)
12. Koenen, R.: MPEG-4 overview. ISO/IEC JTC1/SC29/WG11 document N4668 (March 2002)
13. ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services (2005)
14. Richardson, I.: H.264 and MPEG-4 Video Compression Video Coding for Next-Generation Multimedia. John Wiley & Sons Ltd., England (2003)
15. Wiegand, T., Sullivan, J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. 13, 560–576 (2003)
16. Sullivan, J., Topiwala, P., Luthra, A.: The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions. In: SPIE Conference on Applications of Digital Image Processing XXVII (August 2004)
17. IEEE Standard 1180-1990, IEEE standard specifications for the implementations of 8×8 inverse cosine transform (1990)
18. ISO/IEC IS 23003-1, Information technology – MPEG video technologies – Part 1: Accuracy requirements for implementation of integer-output 8×8 inverse discrete cosine transform (2006)
19. Karczewicz, M., Kurceren, R.: The SP- and SI-frames design for H.264/AVC. IEEE Trans. Circuits Syst. Video Technol. 13, 637–644 (2003)
20. Schwarz, H., Hinz, T., Kirchhoffer, H., Marpe, D., Wiegand, T.: Technical description of the HHI proposal for SVC CE1. ISO/IEC JTC1/SC29/WG11 document M11244 (October 2004)
21. ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services (2007)
22. Wien, M., Schwarz, H., Oelbaum, T.: Performance analysis of SVC. IEEE Trans. Circuits Syst. Video Technol. 17, 1194–1203 (2007)

23. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst. Video Technol. 17, 1103–1120 (2007)
24. MPEG, Call for proposals on multi-view video coding. ISO/IEC JTC1/SC29/WG11 document N7327 (July 2005)
25. MPEG, Requirements on multi-view video coding v.4. ISO/IEC JTC1/SC29/WG11 document N7282 (July 2005)
26. Vetro, A., Pandit, P., Kimata, H., Smolic, A., Wang, Y.-K.: Joint draft 8.0 on multiview video coding. JVT document JVT-AB204 (July 2008)
27. Merkle, P., Smolic, A., Muller, K., Wiegand, T.: Efficient prediction structures for multiview video coding. IEEE Trans. Circuits Syst. Video Technol. 17, 1461–1473 (2007)
28. Reader, C.: AVS intellectual property rights (IPR) policy. J. Comput. Sci. & Techol. 21(3), 306–309 (2006)
29. GB/T20090.2, Information technology – Advanced coding of audio and video – Part2: Video (2006)
30. AVS workgroup, Information technology – Advanced coding of audio and video – Part2: Video. AVS document N1572 (December 2008)
31. AVS workgroup, Information technology – Advanced coding of audio and video – Part2: Video. AVS document N1540 (September 2008)
32. AVS workgroup, Information technology – Advanced coding of audio and video – Part7: Mobility video. AVS document N1151 (December 2004)
33. Zhang, C., et al.: The technique of prescaled integer transform: concept, design and applications. IEEE Trans. Circuits Syst. Video Technol. 18, 84–97 (2008)
34. Wang, Y.-K.: AVS-M: from standards to applications. J. Comput. Sci. & Techol. 21(3), 332–344 (2006)
35. Yi, F., Sun, Q.-C., Dong, J., Yu, L.: Low-complexity tools in AVS part 7. J. Comput. Sci. & Techol. 21(3), 345–353 (2006)
36. MPEG, Vision and requirements for high-performance video coding (HVC). ISO/IEC JTC1/SC29/WG11 document N10361 (February 2009)
37. MPEG, Draft call for evidence on high-performance video coding. ISO/IEC JTC1/SC29/WG11 document N10363 (February 2009)
38. MPEG, Vision on 3D video. ISO/IEC JTC1/SC29/WG11 document N10357 (February 2009)
39. MPEG Introduction to 3D video. ISO/IEC JTC1/SC29/WG11 document N9784 (April 2008)
40. MPEG, Draft call for proposals on reconfigurable video coding (RVC). ISO/IEC JTC1/SC29/WG11 document N7776 (January 2006)
41. MPEG, Whitepaper on reconfigurable video coding (RVC). ISO/IEC JTC1/SC29/WG11 document N9586 (January 2008)