Available online at www.sciencedirect.com
SCIENCE DIRECT®
J. Vis. Commun. Image R. 17 (2006) 186–216
JOURNAL OF VISUAL Communication & IMAGE Representation
www.elsevier.com/locate/jvci

Overview of H.264/MPEG-4 part 10

Soon-kak Kwon [a], A. Tamhankar [b], K.R. Rao [c,*]

[a] *Division of Computer Software Engineering, Dongeui University, Republic of Korea*
[b] *T-Mobile USA, Bellevue, WA, USA*
[c] *Department of Electrical Engineering, University of Texas at Arlington, USA*

Received 16 May 2004; accepted 19 May 2005
Available online 26 August 2005

Abstract

The video coding standards are being developed to satisfy the requirements of applications for various purposes, better picture quality, higher coding efficiency, and more error robustness. The new international video coding standard H.264/MPEG-4 part 10 aims at having significant improvements in coding efficiency, and error robustness in comparison with the previous standards such as MPEG-2, H.263, MPEG-4 part 2. This paper describes an overview of H.264/MPEG-4 part 10. We focus on the detailed features like coding algorithm and error resilience of new standard, and compare the coding schemes with the other standards. The performance comparisons show that H.264 can achieve a coding efficiency improvement of about 1.5 times or greater for each test sequence related to multimedia, SDTV and HDTV.

Keywords: Intra-prediction; Multiple reference Inter-prediction; Integer transform; CAVLC; CABAC

1. Introduction

International study groups, VCEG (Video Coding Experts Group) of ITU-T (International Telecommunication Union—Telecommunication sector), and MPEG

* Corresponding author. Fax: +1 817 272 2253.
E-mail address: rao@uta.edu (K.R. Rao).

1047-3203/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2005.05.010

(Moving Picture Experts Group) [22] of ISO/IEC, have researched the video coding techniques for various applications of moving pictures since the early 1990s. Since then, ITU-T developed H.261 as the first video coding standard for videoconferencing application. MPEG-1 video coding standard was accomplished for storage in compact disk and MPEG-2 [1] (ITU-T adopted it as H.262) standard for digital TV and HDTV as extension of MPEG-1 [16]. Also, for covering the very wide range of applications such as shaped regions of video objects as well as rectangular pictures, MPEG-4 part 2 [2] standard was developed. This includes also natural and synthetic video/audio combinations with interactivity built in. On the other hand, ITU-T developed H.263 [3] in order to improve the compression performance of H.261, and the base coding model of H.263 was adopted as the core of some parts in MPEG-4 part 2. MPEG-1, -2, and -4 also cover audio coding.

To provide better compression of video compared to previous standards, H.264/MPEG-4 part 10 [4] video coding standard was recently developed by the JVT (Joint Video Team) [23] consisting of experts from VCEG and MPEG. H.264 fulfills significant coding efficiency, simple syntax specifications, and seamless integration of video coding into all current protocols and multiplex architectures. Thus, H.264 can support various applications like video broadcasting, video streaming, video conferencing over fixed, and wireless networks and over different transport protocols.

H.264 video coding standard has the same basic functional elements as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261, and H.263) [16], i.e., transform for reduction of spatial correlation, quantization for bitrate control, motion compensated prediction for reduction of temporal correlation, and entropy encoding for reduction of statistical correlation. However, to fulfill better coding performance, the important changes in H.264 occur in the details of each functional element by including intra-picture prediction, a new 4×4 integer transform, multiple reference pictures, variable block sizes and a quarter pel precision for motion compensation, a deblocking filter, and improved entropy coding.

Improved coding efficiency comes at the expense of added complexity to the coder/decoder. H.264 utilizes some methods to reduce the implementation complexity. Multiplier-free integer transform is introduced. Multiplication operation for the exact transform is combined with the multiplication of quantization.

The noisy channel conditions like the wireless networks obstruct the perfect reception of coded video bitstream in the decoder. Incorrect decoding by the lost data degrades the subjective picture quality and propagates to the subsequent blocks or pictures. So, H.264 utilizes some methods to exploit error resilience to network noise. The parameter setting, flexible macroblock ordering, switched slice, redundant slice methods are added to the data partitioning, used in previous standards.

For the particular applications, H.264 defines the Profiles and Levels specifying restrictions on bitstreams like some of the previous video standards. Seven Profiles are defined to cover the various applications from the wireless networks to digital cinema.

Besides H.264, other video coding techniques using the same functional block diagram with some modifications have been developed. These are Mocrosoft Windows

Media Video 9 (WMV-9) [20] by the Society of Motion Picture and Television Engineers (SMPTE) and AVS (Audio Video Coding Standard) [21] by China.

This paper consists of following seven sections as overview of H.264/MPEG-4 part 10. In Sections 2 and 3, Profiles and Levels and the layered structure in bitstream are, respectively, described, Section 4 concentrates on the basic coding algorithms such as intra-prediction, inter-prediction, transform and quantization, entropy coding, B slice, S (switched) slices, and high fidelity coding, Section 5 describes the error resilience methods for transmission error, Section 6 shows the comparisons of H.264 coding schemes with MPEG-2, MPEG-4 part 2, WMV-9, AVS, and Section 7 compares the coding efficiency from the simulated results. Finally, Section 8 presents conclusions.

2. Profiles and Levels

Each Profile specifies a subset of entire bitstream of syntax and limits that shall be supported by all decoders conforming to that Profile. There are three Profiles in the first version: Baseline, Main, and Extended. Baseline Profile is to be applicable to real-time conversational services such as video conferencing and videophone. Main Profile is designed for digital storage media and television broadcasting. Extended Profile is aimed at multimedia services over Internet. Also there are four High Profiles defined in the fidelity range extensions [19] for applications such as content-contribution, content-distribution, and studio editing and post-processing: High, High 10, High 4:2:2, and High 4:4:4. High Profile is to support the 8-bit video with 4:2:0 sampling for applications using high resolution. High 10 Profile is to support the 4:2:0 sampling with up to 10 bits of representation accuracy per sample. High 4:2:2 Profile is to support up to 4:2:2 chroma sampling and up to 10 bits per sample. High 4:4:4 Profile is to support up to 4:4:4 chroma sampling, up to 12 bits per sample, and integer residual color transform for coding RGB signal. The Profiles have both the common coding parts and as well specific coding parts as shown in Fig. 1.

- Common parts of all Profiles
  - I slice (Intra-coded slice): the coded slice by using prediction only from decoded samples within the same slice.
  - P slice (Predictive-coded slice): the coded slice by using inter-prediction from previously decoded reference pictures, using at most one motion vector and reference index to predict the sample values of each block.
  - CAVLC (Context-based Adaptive Variable Length Coding) for entropy coding
- Baseline Profile
  - Flexible macroblock order: macroblocks may not necessarily be in the raster scan order. The map assigns macroblocks to a slice group.
  - Arbitrary slice order: the macroblock address of the first macroblock of a slice of a picture may be smaller than the macroblock address of the first macroblock of some other preceding slice of the same coded picture.

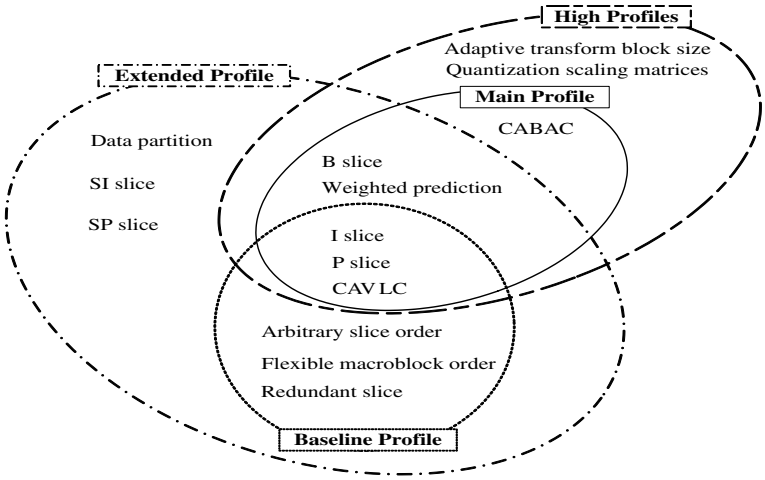


Fig. 1. The specific coding parts of the Profiles in H.264.

  - Redundant slice: this slice belongs to the redundant coded data obtained by same or different coding rate, in comparison with previous coded data of same slice.
- Main Profile
  - B slice (Bi-directionally predictive-coded slice): the coded slice by using inter-prediction from previously decoded reference pictures, using at most two motion vectors and reference indices to predict the sample values of each block.
  - Weighted prediction: scaling operation by applying a weighting factor to the samples of motion-compensated prediction data in P or B slice.
  - CABAC (Context-based Adaptive Binary Arithmetic Coding) for entropy coding
- Extended Profile
  - Includes all parts of Baseline Profile: flexible macroblock order, arbitrary slice order, redundant slice
  - SP slice: the specially coded slice for efficient switching between video streams, similar to coding of a P slice.
  - SI slice: the switched slice, similar to coding of an I slice.
  - Data partition: the coded data is placed in separate data partitions, each partition can be placed in different layer unit.
  - B slice
  - Weighted prediction
- High Profiles
  - Includes all parts of Main Profile: B slice, weighted prediction, CABAC
  - Adaptive transform block size: 4×4 or 8×8 integer transform for luma samples
  - Quantization scaling matrices: different scaling according to specific frequency associated with the transform coefficients in the quantization process to optimize the subjective quality

Table 1
Application requirements [8] (SP, ASP, ARTS, FGS, Studio: Simple, Advanced Simple, Advanced Real Time Simple, Fine Granular Scalability, and Studio Profiles)

Application	Requirements	H.264 Profiles	MPEG-4 Profiles
Broadcast television	Coding efficiency, reliability (over a controlled distribution channel), interlace, low-complexity decoder	Main	ASP
Streaming video	Coding efficiency, reliability (over a uncontrolled packet-based network channel), scalability	Extended	ARTS or FGS
Video storage and Playback	Coding efficiency, interlace, low-complexity encoder and decoder	Main	ASP
Videoconferencing	Coding efficiency, reliability, low latency, low-complexity encoder and decoder	Baseline	SP
Mobile video	Coding efficiency, reliability, low latency, low-complexity encoder and decoder, low power consumption	Baseline	SP
Studio distribution	Lossless or near-lossless, interlace, efficient transcoding	Main High Profiles	Studio

Table 1 lists the H.264 and MPEG-4 part 2 Profiles and important requirements for each application.

For any given Profile, Levels generally correspond to processing power and memory capability of a codec. Each Level may support a different picture size– QCIF, CIF, ITU-R 601 (SDTV), HDTV, S-HDTV, and D-Cinema [16]. Also each Level sets the limits for data bitrate, frame size, picture buffer size, etc.

3. Layered structure

The coded output bitstream of H.264 has two layers, Network Abstraction Layer (NAL) and Video Coding Layer (VCL). NAL abstracts the VCL data in a manner that is appropriate for conveyance on a variety of communication channels or storage media. For the friendliness of communication channel, a NAL unit specifies both byte-stream and packet-based formats. The byte-stream format defines the specific pattern of unique start code prefix for use by applications that deliver some or all of the NAL unit stream as an ordered stream of bytes or bits within the locations of NAL unit boundaries need to be identifiable from patterns in the data, such as H.320 or MPEG-2 system. The packer-based format defines the data packet that are framed by the system transport protocol, without use of start code prefix to prevent the waste of data carrying the prefix, for applications of RTP/UDP/IP. Also, a NAL unit is classified by non-VCL and VCL NAL units. The non-VCL unit contains the additional information such as parameter setting described in Section 5. Previous standards contained header information about slice, picture, sequence that was coded at the start of each element. The loss of packet containing this header information would make the data dependent on this header, as useless. H.264 overcomes this shortcoming by making the packets transmitted synchronously in a real-

time multimedia environment as self-contained [5]. Parameters that change very frequently are added to the slice layer.

The VCL unit contains the core video coded data, which consists of video sequence, picture, slice, and macroblock. The video sequence has either frames or fields which are comprised of three sample arrays, one luma and two chroma sample arrays or the RGB arrays (High 4:4:4 Profile only). Also the standard supports either progressive-scan or interlaced-scan, which may be mixed together in the same sequence. Baseline Profile is limited to progressive scan. Pictures are divided into slices. A slice is a sequence of macroblocks and has the flexible size, especially one slice within a picture. In case of multiple slice groups, the allocation of macroblocks is determined by a macroblock to slice group map that indicates which slice group each MB belongs to. In the 4:2:0 format, each macroblock is comprised of one 16×16 luma and two 8×8 chroma sample arrays. In the 4:2:2 format, the chroma sample arrays are 8×16, and in the 4:4:4, the arrays are 16×16.

4. Video coding algorithm

The block diagram for H.264 coding is shown in Fig. 2. Encoder may select between intra- and inter-coding for block-shaped regions of each picture. Intra-coding can provide access points to the coded sequence where decoding can begin and continue correctly. Intra-coding uses various spatial prediction modes to reduce spatial redundancy in the source signal for a single picture. Inter-coding (predictive or bi-predictive) is more efficient using inter-prediction of each block of sample values from some previously decoded pictures. Inter-coding uses motion vectors for block-based inter-prediction to reduce temporal redundancy among different pictures. Prediction is obtained from deblocking filtered signal of previous reconstructed pictures. The deblocking filter is to reduce the blocking artifacts at the block boundaries. Motion vectors and intra-prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation in the block before it is quantized. Finally, the motion vectors or intra-prediction modes are combined with the quantized transform coefficient information and encoded using entropy code such as context-adaptive variable length codes (CAVLC) or context adaptive binary arithmetic coding (CABAC).

4.1. Intra-prediction

The previous standards have adopted the Intra-coded macroblock, coded by itself without temporal prediction. Intra-coded macroblock occurs in Intra-coded slice or the macroblock having unacceptable temporal correction of motion compensated prediction. Essentially Intra-coded macroblock introduces the high amount of coded bits. This is a bottleneck for reducing the bitrate.

H.264 uses the methods of predicting intra-coded macroblocks to reduce the high amount of bits coded by original input signal itself. For encoding a block or macro-

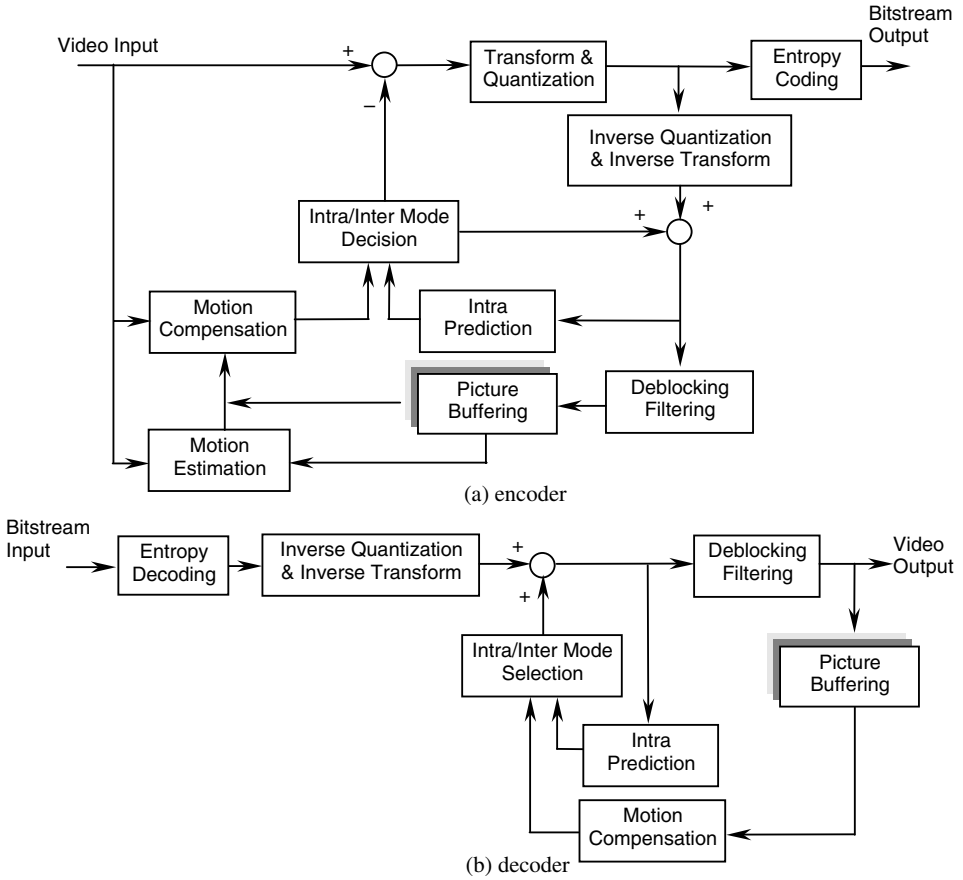


Fig. 2. The block diagram of H.264 algorithm.

block in Intra-coded mode, a prediction block is formed based on previously reconstructed (but, unfiltered for deblocking) blocks. The residual signal between the current block and the prediction is finally encoded. For the luma samples, the prediction block may be formed for each 4×4 subblock, each 8×8 block, or for a 16×16 macroblock. One case is selected from a total of 9 prediction modes for each 4×4 and 8×8 luma blocks; 4 modes for a 16×16 luma block; and 4 modes for each chroma blocks.

Fig. 3 shows a 4×4 luma block that is to be predicted. For the predicted samples $[a, b, \ldots, p]$ for the current block, the above and left previously reconstructed samples $[A, B, \ldots, M]$ are used according to direction modes. The arrows in Fig. 3 indicate the direction of prediction in each mode.

For mode 0 (vertical) and mode 1 (horizontal), the predicted samples are formed by extrapolation from upper samples [A, B, C, D] and from left samples [I, J, K, L],

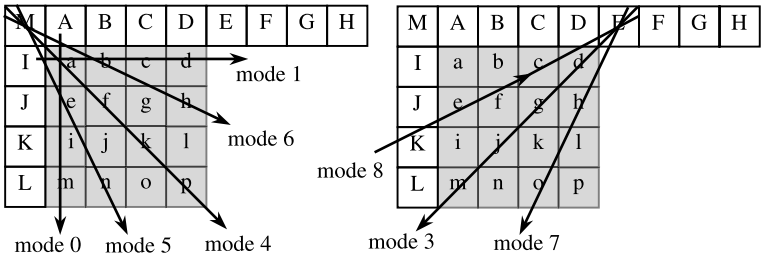


Fig. 3. Intra 4×4 prediction mode directions (vertical, 0; horizontal, 1; DC, 2; diagonal down left, 3; diagonal down right, 4; vertical right, 5; horizontal down, 6; vertical left, 7; and horizontal up, 8).

respectively. For mode 2 (DC), all of the predicted samples are formed by mean of upper and left samples [A, B, C, D, I, J, K, L]. For mode 3 (diagonal down left), mode 4 (diagonal down right), mode 5 (vertical right), mode 6 (horizontal down), mode 7 (vertical left), and mode 8 (horizontal up), the predicted samples are formed from a weighted average of the prediction samples A–M. For example, samples a and d are, respectively, predicted by round(I/4 + M/2 + A/4) and round(B/4 + C/2 + D/4) in mode 4, also by round(I/2 + J/2) and round(J/4 + K/2 + L/4) in mode 8. The encoder may select the prediction mode for each block that minimizes the residual between the block to be encoded and its prediction.

For prediction of each 8×8 luma block, one mode is selected from the 9 modes, similar to the (4×4) intra-block prediction.

For prediction of all 16×16 luma components of a macroblock, four modes are available. For mode 0 (vertical), mode 1 (horizontal), mode 2 (DC), the predictions are similar with the cases of 4×4 luma block. For mode 4 (Plane), a linear plane function is fitted to the upper and left samples.

Each chroma component of a macroblock is predicted from chroma samples above and/or to the left that have previously been encoded and reconstructed. The chroma prediction is defined for three possible block sizes, 8×8 chroma in 4:2:0 format, 8×16 chroma in 4:2:2 format, and 16×16 chroma in 4:4:4 format. The 4 prediction modes for all of these cases are very similar to the 16×16 luma prediction modes, except that the order of mode numbers is different: mode 0 (DC), mode 1 (horizontal), mode 2 (vertical), and mode 3 (plane).

4.2. Inter-prediction

Inter-prediction is to reduce the temporal correlation with help of motion estimation and compensation. In H.264, the current picture can be partitioned into the macroblocks or the smaller blocks. A macroblock of 16×16 luma samples can be partitioned into smaller block sizes up to 4×4. For 16×16 macroblock mode, there are 4 cases: 16×16, 16×8, 8×16 or 8×8, also four cases: 8×8, 8×4, 4×8 or 4×4 for 8×8 mode. The smaller block size requires larger number of bits to signal the motion vectors and extra data of the type of partition, however the motion compensated residual data can be reduced. Therefore, the choice of partition size depends on

the input video characteristics. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

The inter-prediction process can form segmentations for motion representation as small as 4×4 luma samples in size, using motion vector accuracy of one-quarter of the luma sample. Sub-pel motion compensation can provide significantly better compression performance than integer-pel compensation, at the expense of increased complexity. Quarter-pel accuracy outperforms half-pel accuracy. Especially, sub-pel accuracy would increase the coding efficiency at high bitrates and high video resolutions. In the luma component, the sub-pel samples at half-pel positions are generated first and are interpolated from neighboring integer-pel samples using a 6-tap FIR filter with weights (1, −5, 20, 20, −5, 1)/32. Once all the half-pel samples are available, each quarter-pel sample is produced using bilinear interpolation between neighboring half- or integer-pel samples. For 4:2:0 video source sampling, 1/8 pel samples are required in the chroma components (corresponding to 1/4 pel samples in the luma). These samples are interpolated (linear interpolation) between integer-pel chroma samples. Sub-pel motion vectors are encoded differentially with respect to predicted values formed from nearby encoded motion vectors.

The process for inter-prediction of a sample block can also involve the selection of the pictures to be used as the reference pictures from a number of stored previously decoded pictures. Reference pictures for motion compensation are stored in the picture buffer. With respect to the current picture, the pictures before and after the current picture, in the display order are stored into the picture buffer. These are classified as 'short-term' and 'long-term' reference pictures. Long-term reference pictures are introduced to extend the motion search range by using multiple decoded pictures, instead of using just one decoded short-term picture. Memory management is required to take care of marking some stored pictures as 'unused' and deciding which pictures to delete from the buffer for efficient memory management.

4.3. Transform and quantization

Both source pictures and prediction residuals have high spatial redundancies. H.264 Standard is based on the use of a block-based transform for spatial redundancy removal. After inter-prediction from previously decoded samples in other pictures or spatial-based prediction from previously decoded samples within the current picture, the resulting prediction residual is split into 4×4 or 8×8 blocks. These are converted into the transform domain where they are quantized.

H.264 uses an adaptive transform block size, 4×4 and 8×8 (High Profiles only), whereas previous video coding standards used the 8×8 DCT. The smaller block size leads to a significant reduction in ringing artifacts. Also, the 4×4 transform has the additional benefit of removing the need for multiplications.

For improved compression efficiency, H.264 also employs a hierarchical transform structure, in which the DC coefficients of neighboring 4×4 transforms for

the luma signals are grouped into 4×4 blocks and transformed again by the Hadamard transform.

For blocks with mostly flat pel values, there is significant correlation among transform DC coefficients of neighboring blocks. Therefore, the standard specifies the 4×4 Hadamard transform for luma DC coefficients for 16×16 Intra-mode only, and 2×2 Hadamard transform for chroma DC coefficients.

In some applications, it is desired to reduce the quantization step size to improve PSNR to levels that can be considered visually lossless. To achieve this, the H.264 extends the quantization step sizes QP by two additional octaves, redefining the tables and allowing QP to vary from 0 to 51.

In general, transform and quantization require several multiplications resulting in high complexity for implementation. So, for simple implementation, the exact transform process is modified to avoid the multiplications. Then the transform and quantization are combined by the modified integer forward transform, quantization, scaling.

Described below are the steps for the forward integer transform, post-scaling, and quantization for the encoding; and inverse quantization, pre-scaling, and inverse integer transform for the decoding.

- Encoding process

step 1: forward integer transform; For DCT [7] of a 4×4 input luma data F, the exact formula is as follows:

$$X = HFH^{\mathrm{T}}, \tag{1}$$

where, H is a matrix

$$H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}. \tag{2}$$

The variables a, b, c are as follows:

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right), \quad c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right). \tag{3}$$

However, to simplify the implementation of the transform, c is approximated by 0.5 [8,9]. For ensuring orthogonality, b also needs to be modified so that

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{2}{5}}, \quad c = \frac{1}{2}. \tag{4}$$

Multiplication in the transform process is avoided by integrating it with the quantization. So, Eq. (1) is modified as

$$X = \overline{H}F\overline{H}^{\mathrm{T}} \otimes SF, \tag{5}$$

where

$$\overline{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \tag{6}$$

$$SF = \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}. \tag{7}$$

The symbol $\otimes$ denotes the element by element multiplication of the corresponding matrices. In step1, only integer transform is implemented without multiplication term SF. The term SF is combined with quantization in step 2.

step 2: post-scaling and quantization; A transformed and quantized signal Y is obtained by SF ('post-scaling') and quantization step size $Qstep$.

$$Y_{ij} = X_{ij}\text{round}\left(\frac{SF_{ij}}{Qstep}\right). \tag{8}$$

H.264 defines a total of 52 values for $Qstep$.

- Decoding process

step 1: inverse quantization and pre-scaling; A received signal Y in the decoder is scaled with $Qstep$ and SF as the inverse quantization and a part of inverse transform.

$$X'_{ij} = Y_{ij} \bullet Qstep \bullet SF_{ij}^{-1}. \tag{9}$$

step 2: inverse integer transform;

$$F' = \overline{H}_v^{\mathrm{T}} X' \overline{H}_v, \tag{10}$$

where, inverse integer transform matrix is

$$\overline{H}_v = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}. \tag{11}$$

Additionally, for luma (4×4) DC coefficients in 16×16 Intra-mode, 2D Hadamard transform is applied. This is a hierarchical transform

$$\hat{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{12}$$

Also, for chroma DC coefficients in 4:2:0 format, the transform matrix is as follows:

$$\hat{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{13}$$

For 4:2:2 and 4:4:4 formats, the Hadamard block size is increased to reflect the enlarged block.

The following another integer DCT matrix is applied to 8×8 luma components, defined only in High Profiles [19].

$$\overline{H} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}. \tag{14}$$

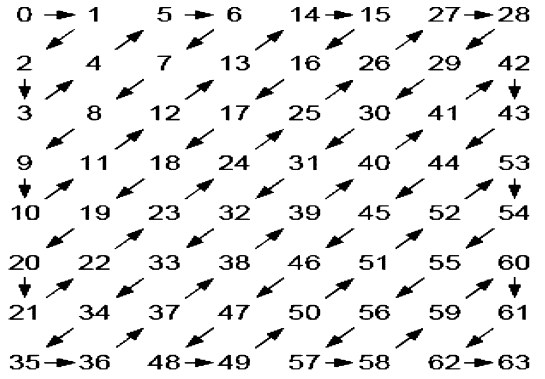
Note that only 4×4 IntDCT is always applied to chroma components.

FRExtensions suggest default perceptual weighting matrices for (4×4) and (8×8) IntDCT coefficients. Scaling matrix reflecting visual perception is simply a multiplier applied during the inverse quantization. (This itself is a multiplication.) Weighting matrices can be customized separately for 4×4 Intra Y, 4×4 Intra Cb, Cr, 4×4 Inter Y, 4×4 Inter Cb, Cr, 8×8 Intra Y, and 8×8 Inter Y.

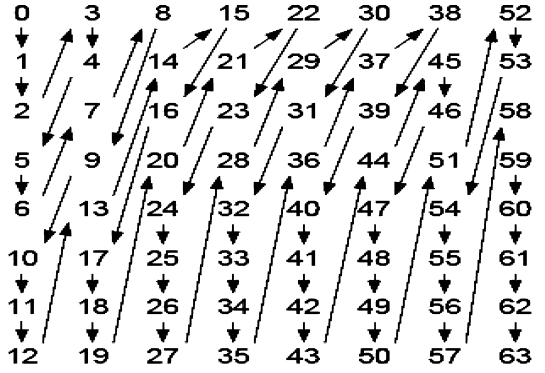
Encoder can design and use customized perceptual scaling matrices. These are to be sent to the decoder at the sequence or picture level. Default scaling matrix for (8×8) IntDCT coefficients is shown below.

$$\begin{bmatrix} 6 & 10 & 13 & 16 & 18 & 23 & 25 & 27 \\ 10 & 11 & 16 & 18 & 23 & 25 & 27 & 29 \\ 13 & 16 & 18 & 23 & 25 & 27 & 29 & 31 \\ 16 & 18 & 23 & 25 & 27 & 29 & 31 & 33 \\ 18 & 23 & 25 & 27 & 29 & 31 & 33 & 36 \\ 23 & 25 & 27 & 29 & 31 & 33 & 36 & 38 \\ 25 & 27 & 29 & 31 & 33 & 36 & 38 & 40 \\ 27 & 29 & 31 & 33 & 36 & 38 & 40 & 42 \end{bmatrix}.$$

In FRExt, two scans similar to 4×4 transform switched for frame/field coding are shown. Coefficient scanning is based on the decreasing variances and to maximize number of zero-valued coefficients along the scan.

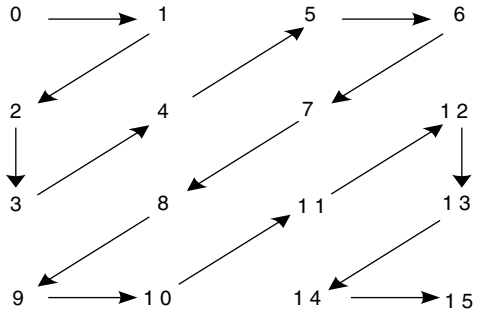


Frame Zig-Zag

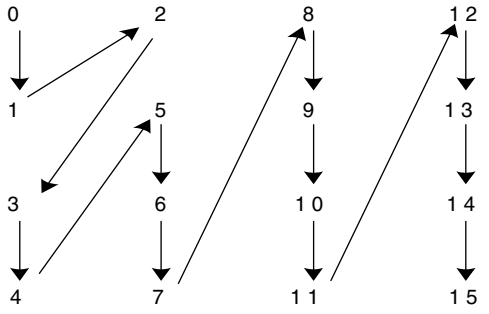


Field

Scanning order of quantized (4×4) IntDCT coefficients is shown below.



Zig-zag scan



Alternate

• Quantization scaling matrices

The High Profiles support the perceptual-based quantization scaling matrices as same concept used in MPEG-2. The encoder can specify a matrix for scaling factor according to the specific frequency associated with the transform coefficient for use in inverse quantization scaling by the decoder. This allows the optimization of the subjective quality according to the sensitivity of the human visual system, less sensitive to the coded error in high frequency transform coefficients.

For the quantization scaling, (8) and (9) are modified as follows:

$$Y_{ij} = X_{ij}\text{round}\left(\frac{SF_{ij}}{Qstep \bullet w_{ij}}\right), \tag{15}$$

$$X'_{ij} = Y_{ij} \bullet Qstep \bullet w_{ij} \bullet SF_{ij}^{-1}, \tag{16}$$

where w_{ij} is a scaling factor.

4.4. Entropy coding

Entropy coding in previous standards such as MPEG-1, -2, -4, H.261, and H.263 is based on fixed tables of variable length codes (VLCs) [16]. These standards define sets of codewords based on the probability distributions of generic videos instead of exact Huffman code for the video sequences. However, H.264 uses different VLCs to match a symbol to a code based on the context characteristics. All syntax elements except for the residual data are encoded by the Exp-Golomb codes [10]. To read the residual data (quantized transform coefficients), zig-zag scan (interlaced) or alternate scan (noninterlaced or field) is used. For coding the residual data, a more sophistical method called CAVLC is employed. Also, CABAC is employed in Main and High Profiles, CABAC has more coding efficiency but higher complexity compared to CAVLC.

4.4.1. Context-based adaptive variable length coding (CAVLC)

After transform and quantization, the probability that the level of coefficients is zero or ± 1 is very high. CAVLC handles the zero and ± 1 coefficients as the different manner with the levels of coefficients. The total numbers of zero and ± 1 are coded. For other coefficients, their levels are coded.

4.4.2. Context-based adaptive binary arithmetic coding (CABAC)

CABAC utilizes the arithmetic coding, also to achieve good compression, the probability model for each symbol element is updated as shown in Fig. 4. The CABAC encoding process consists of three elementary steps [11].

step 1: binarization; A given nonbinary valued symbol (e.g., a transform coefficient or motion vector) is uniquely mapped to a binary sequence prior to arithmetic coding. This process is similar to the process of converting a data symbol into a variable length code but the binary code is further encoded by the arithmetic coder prior to transmission.
step 2: context modeling; A context model is a probability model for one or more elements of the binarized symbol. The probability model is selected such that the corresponding choice may depend on previously encoded syntax elements.
step 3: binary arithmetic coding; An arithmetic coder encodes each element according to the selected probability model together with a subsequent model updating.

4.5. B slice

Bidirectional prediction is very efficient to reduce the temporal correlation by using more reference pictures. Existing standards with B pictures utilize the bidirectional mode, which only allows the combination of a previous and subsequent prediction signals. One prediction signal is derived from the subsequent inter-picture, another from a previous picture, the other from a linear averaged signal of two motion compensated prediction signals.

This H.264 generalizes this concept and supports not only forward/backward prediction pair, but also forward/forward and backward/backward pairs [12]. Two forward references can be beneficial for motion compensated prediction of a region just before scene change, and two backward references just after scene change. In contrast to some other previous standards, bi-directionally predictive-coded slice may also be used as references for inter-coding of other pictures.

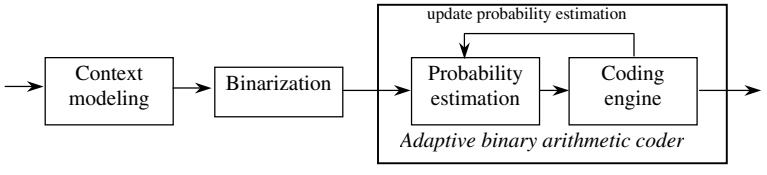


Fig. 4. Schematic block diagram for CABAC.

Also H.264 introduces the direct-mode which does not require such side information but derives reference picture, block size, and motion vector data from the subsequent inter-picture. Weighted prediction is also added for the gradual transitions from scene to scene.

4.5.1. Weighted prediction

All existing standards consider equal weights for reference pictures, i.e., a prediction signal is obtained by averaging with equal weights of reference signals. But, the gradual transitions from scene to scene need the different weights. The gradual transition is very popular in movies, a fade to black scene transition ('fade to black': the luma samples of the scene gradually approach zero and the chroma samples of the scene gradually approach 128), a fade from black scene transition ('fade from black').

H.264 uses weighted prediction method for a macroblock of P slice or B slice. A prediction signal p for B slice is obtained by different weights from two reference signals, $r1$ and $r2$.

$$p = w1 \times r1 + w2 \times r2, \tag{17}$$

where $w1$ and $w2$ are weights. These are differently determined according to two types, explicit and implicit, in encoder. For explicit, the factors are transmitted in the slice header. For implicit, the factors are calculated based on the temporal distance between the pictures. The smaller weight is applied if the temporal distance between the reference and current pictures is close and the larger weight for the temporally long distance.

4.6. Deblocking filter

H.264 may suffer from blocking artifacts due to block-based transform in intra- and inter-prediction coding, and the quantization of the transform coefficients. The deblocking filter reduces the blocking artifacts in the block boundary and prevents the propagation of accumulated coded noise. H.261 has suggested similar deblocking filter (optional) which was beneficial to reduce the temporal propagation of coded noise because only integer-pel accuracy motion compensation did not play the role for its reduction. However, MPEG-1, -2 did not use the deblocking filter because of high implementation complexity, on the other hand, the blocking artifacts can be reduced by utilizing the half-pel accuracy MC. The half-pels obtained by bilinear filtering of neighboring integer-pels played the role of the smoothing of the coded noise in the integer-pel domain.

H.264 uses the deblocking filter for higher coding performance in spite of implementation complexity as shown in Fig. 5. Filtering is applied to horizontal or vertical edges of 4×4 blocks in a macroblock. The luma deblocking filter process is performed on four 16-sample edges and the deblocking filter process for each chroma components is performed on two 8-sample edges.

The deblocking filter is applied adaptively at several levels [6].

- slice level: the global filtering strength can be adjusted to the individual characteristics of the video sequence.

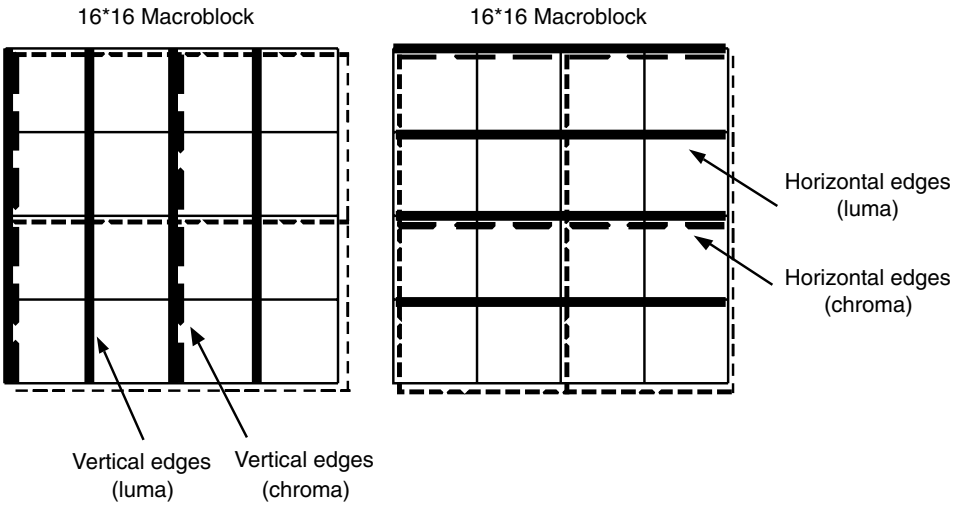


Fig. 5. Boundaries in a macroblock to be filtered (luma boundaries shown with solid lines and chroma boundaries shown with dotted lines) [4].

- block-edge level: filtering strength is made dependent on the inter-/intra-prediction decision, motion differences, and the presence of coded residuals in the two participating blocks. Special strong filtering is applied for macroblocks with very flat characteristics to remove 'tilting artifacts'
- sample level: sample values and quantizer-dependent thresholds can turn off filtering for each individual sample.

4.7. SP and SI slices

In the previous standards, perfect switching between bitstreams is possible only at I picture. Reconstructing I pictures at fixed intervals allows the random access or fast playback. However, the drawback of using I picture is that it requires large number of bits, since I pictures do not exploit any temporal redundancies.

H.264 introduces the switched slices, SP and SI, for bitstream switching [13]. Fig. 6 shows an example how to utilize SP pictures to switch between different bitstreams. Assume that there are two bitstreams, $P(1,k)$ and $P(2,k)$, corresponding to the same sequence encoded at different bit rates. Within each encoded bitstream, SP-pictures are placed at the locations at which switching from one bitstream to another will be allowed.

In the case of switching from above bitstream $P(1,3)$ to $P(2,3)$, a SP picture $S(3)$ allows to produce the decoded picture $P(2,3)$ by using $P(1,2)$ in the other bitstream even though motion compensations are included.

SI slice, is used in a way similar to SP slice, but prediction is formed by using the 4×4 intra-prediction modes from previously decoded samples of the reconstructed picture.

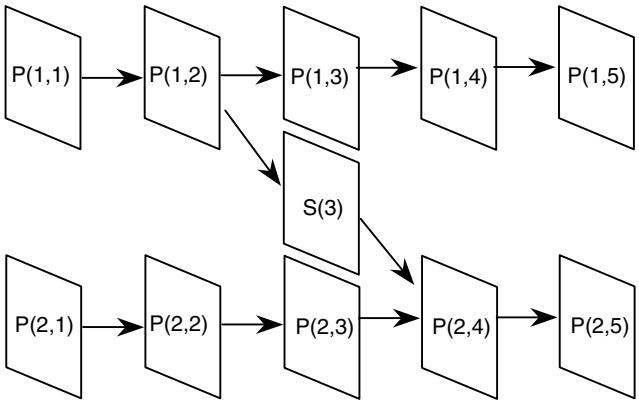


Fig. 6. Switching using SP picture.

4.8. High fidelity coding

H.264 defines the special coding schemes for fidelity range extension in High Profiles [19]. There are lossless coding schemes and support of several color formats.

- Lossless coding;

To represent the video signal as high fidelity, H.264 specifies the lossless coding schemes in High 4:4:4 Profile only. First one is PCM scheme, in which the values of the samples are sent directly for perfectly lossless representation—without prediction, transformation, or quantization. Second one is transform-bypass lossless coding scheme, which uses prediction and entropy coding of encoding sample values for fairly efficient lossless representation. Second scheme introduces less coded data than first one.

- Support of several color formats;

By performing color transformation of RGB-to-YCbCr, rounding error is introduced in both the forward and inverse color transformations. Therefore, to eliminate the rounding error induced in floating point operations, H.264 adds support for a new color space YCgCo as well as RGB in High Profiles [19].

$$Y = \frac{1}{2}\left(G + \frac{R+B}{2}\right), \quad Cg = \frac{1}{2}\left(G - \frac{R+B}{2}\right), \quad Co = \frac{R-B}{2}. \tag{18}$$

Above equations reduce the complexity in the transformation but require the additional bits of accuracy to avoid the rounding errors. To reduce the bit expansion of sample accuracy to 1 bit, the following equations are used:

$$Co = R - B, \quad Cg = G - (B + Co \gg 1), \quad Y = (B + Co \gg 1) + (Cg \gg 1). \tag{19}$$

5. Error resilience

The previous standards use some methods to exploit the error resilience from transmission noise. The data partitioning are popular. The data are partitioned

according to the significance in the bitstream, the important data then are transmitted with high priority. Also, the layered (scalable) coding induces the error resilience. Spatial or temporal scalable coding can recover the lost data from other layers.

H.264 also has the error resilience property with help of S slice, parameter setting, flexible macroblock ordering, and redundant slice [14].

• Parameter setting;

The sequence parameter set contains all information related to a sequence of pictures and a picture parameter set contains all information related to all the slices belonging to a single picture. The encoder chooses the appropriate picture parameter set to use by referencing the storage location in the slice header of each coded slice. The intelligent use of the parameter set mechanism greatly enhances error resilience. The key to using parameter sets in an error-prone environment is to ensure that they arrive reliably, and in a timely fashion at the decoder. They can, for example, be sent out-of-band as shown in Fig. 7, using a reliable control protocol, so that the control protocol time to get them to the decoder before the relevant slices arrive over the real-time communication channel.

Alternatively, they can be sent in-band, but with appropriate application layer protection (e.g., by sending multiple copies, so as to enhance the probability that at least one copy arrives at the destination). A third option is that an application hard-codes a few parameter sets in both encoder and decoder, which would be the only operation points of the codec.

• Flexible macroblock ordering;

Flexible macroblock ordering allows assigning macroblocks to slices in an order other than the scan order. To do so, each macroblock is statically assigned to a slice group using a macroblock allocation map. Within a slice group, macroblocks are coded using the normal scan order. Assume that all macroblocks of the picture are allocated either to slice group 0 or slice group 1, and the macroblocks in each

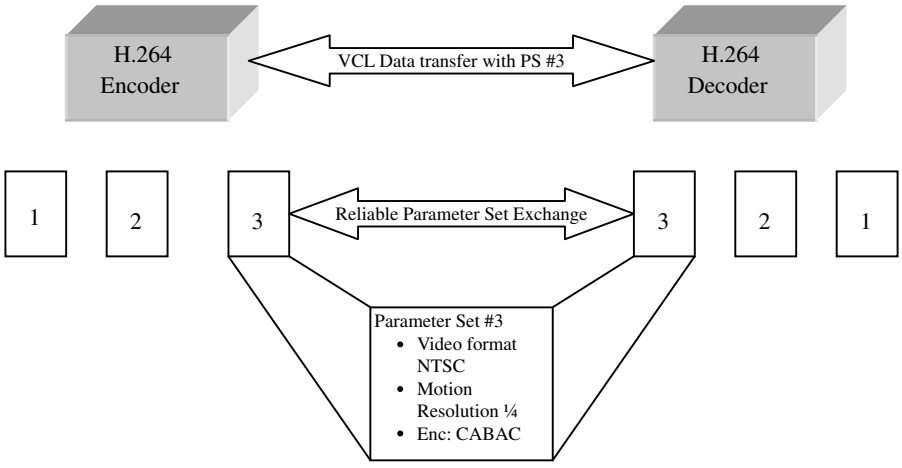


Fig. 7. Parameter setting.

slice group are dispersed throughout the picture. If the packet containing the information of slice group 1 is lost during transmission, then the lost macroblock can be recovered by the error concealment mechanism, since every lost macroblock has several spatial neighbors that belong to the other slice.

- Redundant slice;

Redundant slices allow to place one or more redundant representations of the same macroblocks into the same bitstream, in addition to the coded macroblocks of the slice itself. The redundant representation can be coded using different coding parameters.

For example, the primary representation can be coded with a low quantization parameter (hence in good quality), whereas the redundant slice can be coded with a high quantization parameter (hence, in a much coarser quality, but also utilizing fewer bits). A decoder reacts to redundant slices by reconstructing only the primary slice, if it is available, and discarding the redundant slice. However, if the primary slice is missing, the redundant slice can be reconstructed.

6. Comparison of coding schemes with other standards

First, we compare the coding algorithms of H.264 with MPEG-2 and MPEG-4 part 2 (Table 2).

New functionalities introduced in H.264 are intra-prediction in the spatial domain, hierarchical transform with $(4 \times 4, 8 \times 8)$ integer DCT transforms and (2 times $2, 4 \times 4$) Hadamard transforms, multiple reference pictures in inter-prediction, generalized bidirectional prediction (forward/forward, backward/backward), weighted prediction, deblocking filter, CAVLC and CABAC entropy coding, parameter setting, flexible macroblock ordering, redundant slice, and SP/SI slices for error resilience.

Second, other coding standards such as Windows Media Video 9 (WMV-9) [20] and AVS China [21] are briefly compared with H.264 in Table 3.

- Windows Media Video 9 [20]:

WMV-9 takes adaptive block size transform which allows 8×8 blocks to be encoded using either one 8×8, two horizontally stacked 8×4 s, two vertically stacked 4×8 s, or four 4×4 block transforms. Also the integer transforms are used but these matrixes are different compared with H.264. A matrix for 8×8 inverse transform is as follows:

$$\begin{bmatrix} 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 16 & 15 & 9 & 4 & -4 & -9 & -15 & -16 \\ 16 & 6 & -6 & -16 & -16 & -6 & 6 & 16 \\ 15 & -4 & -16 & -9 & 9 & 16 & 4 & -15 \\ 12 & -12 & -12 & 12 & 12 & -12 & -12 & 12 \\ 9 & -16 & 4 & 15 & -15 & -4 & 16 & -9 \\ 6 & -16 & 16 & -6 & -6 & 16 & -16 & 6 \\ 4 & -9 & 15 & -16 & 16 & -15 & 9 & -4 \end{bmatrix}.$$

Table 2
Comparison of standards MPEG-2, MPEG-4 part 2, and H.264/MPEG-4 part 10

Feature/standard	MPEG-2	MPEG-4 part 2	MPEG-4 part 10 /H.264
Macroblock size	16×16 (frame mode), 16×8 (field mode)	16×16	16×16
Block size	8×8	16×16, 16×8, 8×8	16×16, 8x16, 16×8, 8×8, 4×8, 8×4, 4×4
Intra-prediction	No	Transform domain	Spatial domain
Transform	8×8 DCT	8×8 DCT/Wavelet transform	8×8, 4×4 integer DCT, 4×4, 2×2 Hadamard
Quantization	Scalar quantization with step size of constant increment	Vector quantization	Scalar quantization with step size of increase at the rate of 12.5%
Entropy coding	VLC	VLC	VLC, CAVLC, CABAC
Pel accuracy	1/2-pel	1/4-pel	1/4-pel
Reference picture	One picture	One picture	Multiple pictures
Bidirectional prediction mode	Forward/backward	Forward/backward	Forward/backward, forward/forward, backward/backward
Weighted prediction	No	No	Yes
Deblocking filter	No	No	Yes
Picture types	I, P, B	I, P, B	I, P, B, SI, SP
Profiles	5 profiles	8 profiles	7 profiles
Playback and random access	Yes	Yes	Yes
Error robustness	Data partitioning, FEC for important packet transmission	Synchronization, data partitioning, header extension, reversible VLCs	Data partitioning, parameter setting, flexible macroblock ordering, redundant slice, SP and SI slices
Transmission rate	2–15 Mbps	64 kbps–2 Mbps	64 kbps–150 Mbps
Encoder complexity	Medium	Medium	High
Compatibility with previous standards	Yes	Yes	No

Also, a matrix for 4×4 inverse transform is as follows:

$$\begin{bmatrix} 17 & 17 & 17 & 17 \\ 22 & 10 & -10 & -22 \\ 17 & -17 & -17 & 17 \\ 10 & -22 & 22 & -10 \end{bmatrix}.$$

WMV-9 allows both dead-zone and regular uniform quantization, a quantization parameter-based rule allows for an automatic switch from regular uniform quantization to dead-zone uniform quantization as the parameter increases. Simple variable length codes are used for entropy coding, but the use of multiple code tables for encoding each symbol is allowed.

For sub-pel motion vector accuracy, the four-tap bicubic filters is used, $[-4\ 53\ 18\ -3]/64$ for 1/4 pel accuracy and $[-1\ 9\ 9\ -1]/16$ for 1/2 pel accuracy.

Table 3
Comparison of standards H.264/MPEG-4 part 10, WMV-9, and AVS

Feature/standard	MPEG-4 part 10/H.264	WMV-9	AVS
Prediction block size	16×16, 8×16, 16×8, 8×8, 4×8, 8×4, 4×4	16×16, 8×8	16×16, 8×8
Intra-prediction	4×4, 8×8: 9 modes 16×16: 4 modes	No	8×8: 5 modes
Transform	8×8, 4×4 integer DCT 4×4, 2×2 Hadamard	8×8, 8×4, 4×8, 4×4 integer DCT	Asymmetric 8×8 integer DCT
Quantization	Scalar quantization	Dead zone, uniform scalar quantization	Scalar quantization
Entropy coding	VLC, CAVLC, CABAC	Multiple VLC tables	VLC
Sub-pel filter	1/2-pel: 6-tap, 1/4-pel: 2-tap	1/2-pel: 4-tap, 1/4-pel: 4-tap	1/2-pel: 4-tap, 1/4-pel: 4-tap
Reference picture	Multiple pictures	One picture	Two pictures
Bidirectional prediction mode	Forward/backward, forward/forward, backward/backward, 2 motion vectors	Forward/backward, 2 motion vectors	Forward/backward, symmetric 1 motion vector
Weighted prediction	Yes	Yes	Yes
Deblocking filter	Yes	Yes	Yes

For bidirectional prediction, timing for scaling of motion vectors is explicitly included in direct mode. Intra-coded B pictures are allowed for scene change. Bottom B-fields can be referred to top fields from the same picture in interlace coding. Similar deblocking filter is used to remove the block-boundary discontinuities and the weighted prediction is used to improve the performance of motion compensation on video sequences that include fading.

WMV-9 introduces the special techniques, overlapped transform and low-rate tool. To minimize the blocking effect, cross-block correlations can be exploited by means of a overlapped transform. A overlapped transform is a transform whose input spans, besides the data elements in the current block, a few adjacent elements in neighboring blocks. Low-rate tool is the ability to code pictures at multiple resolutions, by scaling down the both horizontal and vertical directions of each coded picture. The decoder is informed that these pictures have been scaled down, and it will up-scale the decoded image in both directions as appropriate before displaying them.

• AVS: The Chinese Next-Generation Video Coding Standard [21]:

AVS is focused on the applications of broadcast TV, HD-DVD, and broadband video networking and mobile networks. AVS allows the 8×8 block size only.

For intra-prediction, one case is selected from 5 modes (mode 0, vertical; mode 1, horizontal; mode 2, DC; mode 3, diagonal down left; and mode 4, diagonal down right), but DC is obtained after lowpass filtering.

AVS uses a separable, integer-precise, 8×8 DCT and asymmetric transform in which both post-scaling and pre-scaling are involved in encoder side. A matrix for 8×8 inverse transform is as follows:

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

Quantization of the transform coefficients is performed with a linear scalar quantizer.

For entropy coding, all syntax elements including transform coefficients are encoded by the Exp-Golomb codes. For motion vector accuracy, sub-pels are interpolated by a process in which the values at 1/2 pel locations are calculated using a 4-tap filter: [−1 5 5 −1]/8, and the values at 1/4 pel locations are calculated using a 4-tap filter [1 3 3 1]/8.

Inter-prediction for P picture may be from two reference pictures, the most recent picture or the second most recent picture. The prediction for B picture is obtained by the average of signals in the most recent and future P/I pictures as in MPEG-2. Direct mode is used as in H.264, but new symmetric mode is introduced, in which a single motion vector is calculated and transmitted that passes through the macroblock in the interpolated picture, pointing to the past and future P pictures. Also, similar deblocking filter and weighted prediction are used.

7. Comparison of coding efficiency

Several papers [6,9,11,12,17,18] compared H.264 coding efficiency with existing standards. Also, JVT simulated the coding efficiency of H.264 standard and compared with MPEG-2 and MPEG-4 part 2 visual by test software (H.264/AVC JM software can be downloaded from [26]). Tables 4–7 are from JVT document [15]. In the simulation, the formal subjective verification tests have been statistically processed to obtain the Mean Opinion Score (MOS), calculated by averaging the opinions of the subjects.

Table 4 shows the comparison of the H.264 Baseline Profile (BP) and MPEG-4 part 2 Simple Profile (SP), in which the test sequences have the multimedia definition (MD). The numbers in the table indicate the coding efficiency improvement achieved

Table 4
Comparison of H.264 BP and MPEG-4 part 2 SP for the MD Baseline test

Sequence	Bitrate [kbps] for QCIF				Bitrate [kbps] for CIF			
	24	48	96	192	96	192	384	768
Foreman	>1×	2×	2×	T	2×	>2×	T	T
Paris	>1×	2×	2×	—	2×	2×	T, 2×	T
Head		>2×	2×	—	2×	—	T	T
Zoom	>1×	1×	2×	—	2×	—	—	—

Table 5
Comparison of H.264 MP and MPEG-4 part 2 ASP for the MD Main test

Sequence	Bitrate [kbps] for QCIF				Bitrate [kbps] for CIF			
	24	48	96	192	96	192	384	768
Football	2×/1×	2×	2×	—	>1×	>1×	1×	>1×
Mobile	2×/1×	2×	2×	—	>2×	4×	>2×	T
Husky	2×	2×	>1×	—	2×	2×	2×	—
Tempete	2×	2×	>2×	T	2×	2×	T, 2×	T

Table 6
Comparison of H.264 MP and MPEG-2 for the SD Main test

Sequence	Bitrate [Mbps] for MPEG-2 *HiQ*					Bitrate [Mbps] for MPEG-2 TM5				
	1.5	2.25	3	4	6	1.5	2.25	3	4	6
Football	>1.5×	>1.3×	1.3×	1.5×	—	2×	1.8×	1.3×	1.5×	—
Mobile	4×	2.7×	2×	T	T	>4×	>2.7×	>2×	T	T
Husky	>1.5×	1.3×	1×/1.3×	1.5×	—	2.7×/2×	1.8×	2×	>1.5×	—
Tempete	T, 2×	T	T	T	T	T, 4×	T	T	T	T

Table 7
Comparison of H.264 MP and MPEG-2 for the HD Main test

Sequence		Bitrate [Mbps] for MPEG-2 *HiQ*			Bitrate [Mbps] for MPEG-2 TM5		
		6	10	20	6	10	20
720 (60p)	Crew	1.7×	2×	T	1.7×	2×	T
	Harbour	T, 3.3×	T	T	T, 1.7×	T	T
1080 (30i)	Stockholm Pan	—	1×	—	—	2×	—
	New mobile and calendar	—	T, 2×	T	—	T, 2×	T
1080 (25p)	River bed	>1.7×	>1×	T	>1.7×	>1×	T
	Vintage car	1.7×	T, 2×	T	1.7×	T, 2×	T

by the H.264 codec where the codecs based on other standards being compared provide statistically equivalent picture quality. To obtain the bitrates at which the other codecs are using to achieve the same quality as the H.264, one needs to multiply this bitrate by the $N\times$ value found in the table. The letter 'T' indicates that H.264 achieved transparency at the bitrate for the given sequence. H.264 Baseline Profile achieved a coding efficiency improvement of two times or greater in 14 out of 18 statistically conclusive cases. The '—' indicates statistically inconclusive test.

Table 5 shows the comparison of H.264 Main Profile (MP) and MPEG-4 part 2 Advanced Simple Profile (ASP) [2] for the MD. H.264 Main Profile achieved a coding efficiency improvement of two times or greater in 18 out of 25 statistically conclusive cases.

Table 6 shows the comparison of H.264 Main Profile and MPEG-2 for the standard definition (SD). When compared to MPEG-2 HiQ (real-time High Quality) [1], H.264 Main Profile achieved a coding efficiency improvement of 1.5 times or greater in 8 out of 12 statistically conclusive cases, out of which three cases show improvements of two times or greater and in 1 case shows an improvement of four times.

When compared to MPEG-2 TM5, H.264 Main Profile achieved a coding efficiency improvement of 1.8 times or greater in 9 out of 12 statistically conclusive cases, out of which two cases show improvements of four times or greater.

Table 7 shows the comparison of H.264 Main Profile and MPEG-2 for the high definition (HD). When compared to MPEG-2 HiQ, H.264 Main Profile achieved a coding efficiency improvement of 1.7 times or greater in 7 out of 9 statistically conclusive cases, out of which three cases show improvements of two times or greater and in 1 case shows an improvement of 3.3 times.

When compared to MPEG-2 TM5, H.264 Main Profile achieved a coding efficiency improvement of 1.7 times or greater in 8 out of 9 statistically conclusive cases, out of which four cases show improvements of two times or greater.

Also, the results of objective tests are shown in Figs. 8 and 9 [17]. In the simulation, full search motion estimation with a range of 32 integer pixels was used by all encoders along with the Lagrangian Coder described in [17].

Fig. 8 shows the PSNR (between original and reconstructed pixels) and bitrate savings for 'Tempete' CIF 15 Hz sequence for the video streaming application. For this experiment, MPEG-2 with Main Profile and Main Level (MP@ML), H.263 of High-Latency Profile (HLP), MPEG-4 part 2 with Advanced Simple Profile (ASP), and H.264/MPEG-4 part 10 Main Profile (H.26L) were used. Also, five reference pictures were used for both H.263 and H.264/MPEG-4 part 10.

We can see that H.264 improves the coding efficiency of about 3.6 dB over MPEG-2 MP@ML, about 2.8 dB over H.263 HLP, and about 2.2 dB over MPEG-4 ASP at 512 kbit/s. Also H.264 achieves the bitrate savings of 50–70%, when compared with MPEG-2 MP@ML.

Fig. 9 shows the PSNR and bitrate saving results for 'Paris' CIF 15 Hz sequence for the video conferencing application. For this experiment, H.263 Baseline, Conversational High Compression (CHC), MPEG-4 SP, ASP, and H.264/MPEG-4 part 10 Baseline Profile (H.26L) were used. H.264 improves the coding efficiency by more than 2 dB compared to the other standards.

7.1. Performance of FRExt High Profile

The subjective quality evaluation done by the Blu-ray Disc Association (BDA) is shown in Fig. 10 [27]. This test, conducted on 24 frame/s film content with 1920×1080 progressive-scanning, shows the following nominal results (which may or may not be rigorously statistically proven):

- The High Profile of FRExt produced nominally *better* video quality than MPEG-2 when using only *onethird* as many bits (8 Mbps versus 24 Mbps).
- The High Profile of FRExt produced nominally *transparent* (i.e., difficult to distinguish from the original video without compression) video quality at only 16 Mbps.

The quality bar (3.0), considered adequate for use on high-definition packaged media in this organization, was significantly surpassed using only 8 Mbps.

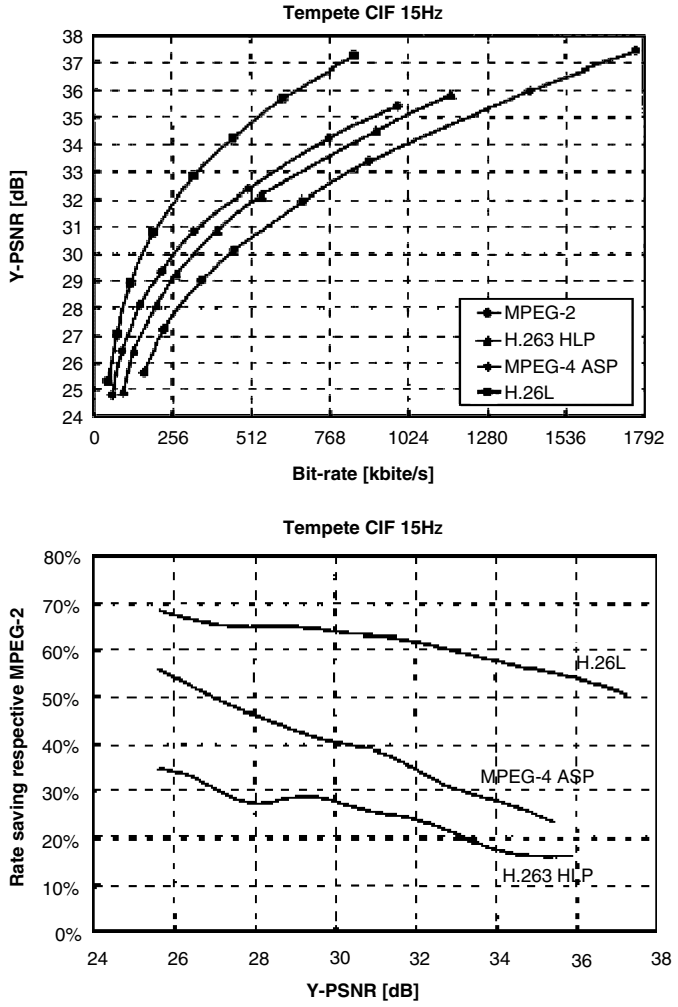


Fig. 8. Rate distortion and bitrate saving curves for 'Tempete' CIF 15 Hz sequence [17] (HLP, High Latency Profile; ASP, Advanced Simple Profile; and H.26L, H.264 Main Profile).

PSNR comparison test performed by FastVDO is shown in Fig. 11. These objective results confirm the strong performance of the High Profile. (Sub-optimal uses of B frames and other aspects make the plotted performance conservative for FREXt, thus the remark in the figure about possible future performance.)

8. Audio coding and systems

While the scope of H.264/MPEG-4 part 10 is limited to video, audio is required for all practical applications. Specific audio coding algorithm, bit rates, number of

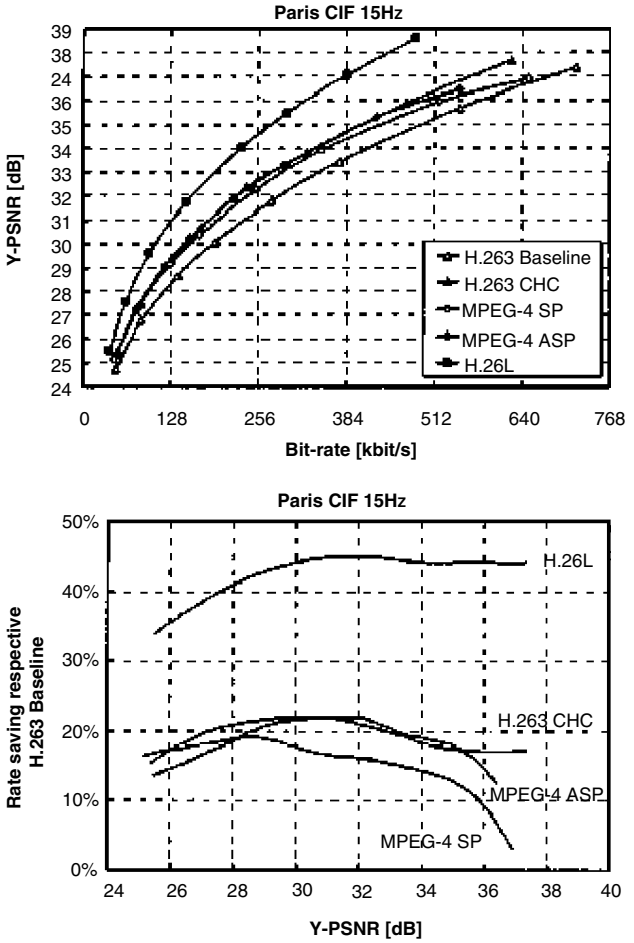


Fig. 9. Rate-distortion and bit-rate saving curves for 'Paris' CIF 15 Hz sequence [17] (CHC, Conversational High Compression; SP, Simple Profile; ASP, Advanced Simple Profile; and H.26L, H.264 Baseline Profile).

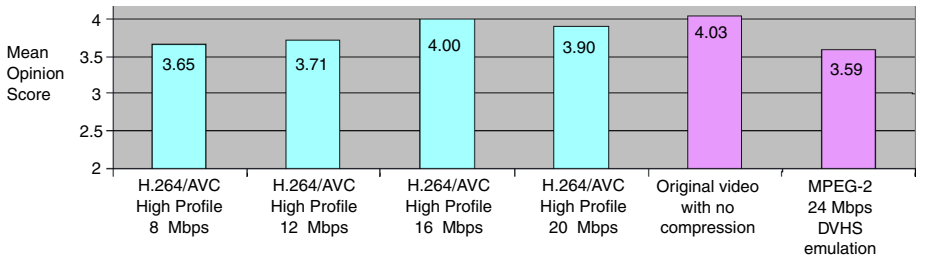


Fig. 10. Perceptual test of FREXt High Profile capability by Blu-ray Disc Association [27].

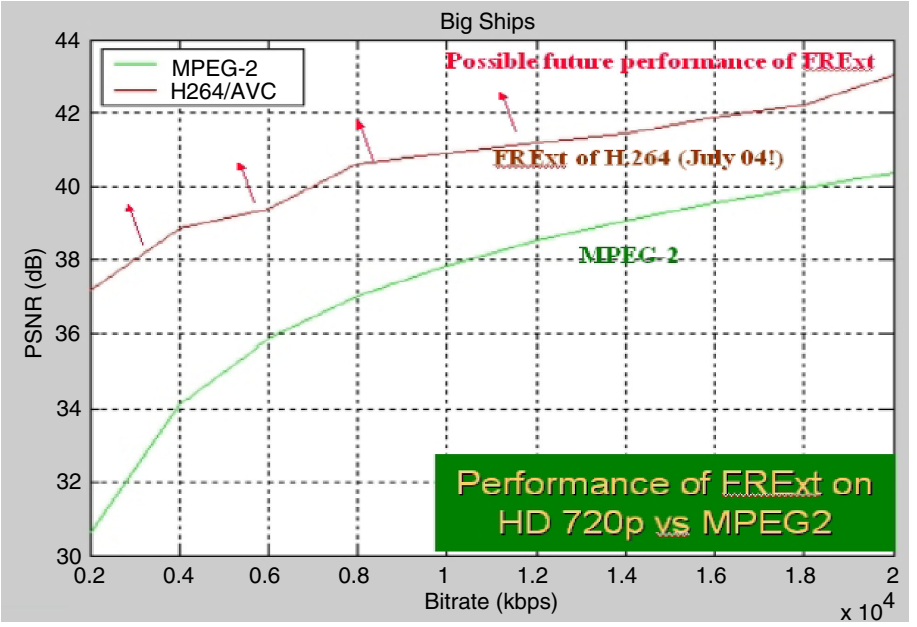


Fig. 11. Objective performance of FREXt High Profile vs. MPEG-2 on a test 720p clip. These results are consistent with the subjective results in Fig. 10, i.e, 8 Mb/s of FREXt outperforms 20 Mb/s of MPEG-2.

channels (mono, stereo, surround sound, etc.) and quality levels are left to the industry and organizations such as ATSC (US terrestrial broadcast0, SCTE US/CANADA cable), ARIB (Japan), and DVB (Europe). DVB is considering AAC with SBR, called AAC plus (AAC, advanced audio coder; SBR, spectral band replication) [47–51] while ATSC has selected AC-3 plus from Dolby Labs; MPEG calls it as HE-AAC (HE, high efficiency). In addition for compatibility all the application standards (ATSC, SCTE, ARIB, DVB, etc.) will continue to use the existing standards (MPEG-1 audio, AC-3, and MPEG-2 AAC). The glue to all these is the MPEG-2 transport stream that provides the audio/video synchronization mechanism for all the audio/video standards.

9. WMV9 and AVS China

Both WMV9 [20] (windows media video 9 from Microsoft) and AVS China [21] (AVS—audio video standard from china) have developed encoder/decoder algorithms similar to H.264. WMV9, AVS China, and H.264 have similar functionalities with some changes in adaptive/non-adaptive transform sizes, entropy coding, directional prediction, interpolation filters for fractional pel ME, role of B frames, deblocking filter, etc. Both AVS China and WMV9 have flexibility in bit rates, quality levels, spatial/temporal resolutions, color formats, etc., and address a spectrum of applications.

10. Systems

ITU-T systems (H.32X), MPEG-4 systems/file format, and H.322/MPEG2 systems are modified to support H.264 video and audio bit streams, i.e., MPEG-2 TS (transport stream) amendment 3 facilitates transport of MPEG-4 AVC. Conformance bitstream (H.264.1) and reference software (H.264.2) have been approved.

11. Conclusions

This paper summarizes the recent video coding standard, H.264/MPEG-4 part 10, developed by the Joint Video Team of ITU-T and MPEG. H.264 outperforms over the previous standards by introducing the special coding algorithms such as intra-prediction, 4×4 integer transform, several block sizes, quarter-pel accuracy motion vector, and multiple reference prediction, and weighted prediction for motion compensation, deblocking filter, CAVLC, and CABAC. Also, for error resilience, parameter setting, flexible macroblock ordering, redundant slice, and SP and SI slices are employed.

Currently, the commercial H.264 codecs [28–46] are widely developed by several companies for replacing/complementing existing products. This is only a partial list. H.264 can be applied to consumer equipment such as digital cameras, cell phones, video-over-IP networks, and high-definition digital broadcasting through terrestrial or satellite channels, and digital storage system such as high-definition DVD.

However, the commercial implementation of H.264 will be inside of the scope of various patents. MPEG LA [24] and via licensing [25] now coordinating the licensing terms, decoder-encoder royalties for product manufacturers and participation fees for video streaming services regardless of Profile(s).

Acknowledgment

S.K. Kwon acknowledges support by the Post-doctoral Fellowship Program of Korea Science and Engineering Foundation.

References

[1] MPEG-2: ISO/IEC JTC1/SC29/WG11 and ITU-T, ISO/IEC 13818-2: Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC and ITU-T, 1994.

[2] MPEG-4: ISO/IEC JTCI/SC29/WG11, ISO/IEC 14 496:2000-2: Information on Technology-Coding of Audio-Visual Objects-Part 2: Visual, ISO/IEC, 2000.

[3] H.263: International Telecommunication Union, Recommendation ITU-T H.263: Video Coding for Low Bit Rate Communication, ITU-T, 1998.

[4] H.264: International Telecommunication Union, Recommendation ITU-T H.264: Advanced Video Coding for Generic Audiovisual Services, ITU-T, 2003.

[5] T. Stockhammer, M. Hannuksela, S. Wenger, H.26L/JVT coding network abstraction layer and IP-based transportIEEE ICIP 2002, vol. 2, Rochester, New York, 2002, pp. 485–488.

[6] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, Adaptive deblocking filter, IEEE Trans. CSVT 13 (2003) 614–619.
[7] K.R. Rao, P. Yip, Discrete Cosine Transform, Academic Press, 1990.
[8] I.E.G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next-generation, Wiley, 2003.
[9] H.S. Malvar et al., Low-complexity transform and quantization in H.264/AVC, IEEE Trans. CSVT 13 (2003) 598–603.
[10] S.W. Golomb, Run-Length Encoding, IEEE Trans. Informat. Theory, IT-12, (1966) 399–401.
[11] D. Marpe, H. Schwarz, T. Wiegand, Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard, IEEE Trans. CSVT 13 (2003) 620–636.
[12] M. Flierl, B. Girod, Generalized B picture and the draft H.264/AVC video-compression standard, IEEE Trans. CSVT 13 (2003) 587–597.
[13] M. Karczewicz, R. Kurceren, The SP- and SI-frames design for H.264/AVC, IEEE Trans. CSVT 13 (2003) 637–644.
[14] S. Wenger, H.264/AVC over IP, IEEE Trans. CSVT 13 (2003) 645–656.
[15] ISO/IEC JTC1/SC29/WG11, Report of The Formal Verification Tests on AVC (ISO/IEC14496-10 | ITU-T Rec. H.264), MPEG2003/N6231, December 2003.
[16] M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, IEE, Hertz, UK, 2003.
[17] A. Joch et al., Performance comparison of video coding standards using lagrangian coder control. IEEE ICIP 2002, vol. 2, Rochester, New York, 2002, pp. 501–504.
[18] T. Wiegand et al., Overview of the H.264/AVC video coding standard, IEEE Trans. CSVT 13 (2003) 560–576.
[19] G. Sullivan, P. Topiwala, A. Luthra, 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, 2004.
[20] S. Srinivasan et al., Windows media video 9: overview and applications, Signal Process.: Image Commun. 19 (2004) 851–875.
[21] W. Gao et al., AVS—The Chinese next-generation video coding standard, NAB 2004, Las Vegas, 2004.
[22] MPEG website: http://www.mpeg.org.
[23] JVT website: ftp://standards.polycom.com.
[24] MPEG LA website: http://www.mpegla.com.
[25] Via licensing: www.vialicensing.com.
[26] H.264/AVC JM Software: http://bs.hhi.de/~ suehring/tml/download new version of the H.264/AVC reference software http://iphome.hhi.de/suehring/tml/download/.
[27] T. Wedi, Y. Kashiwagi, Subjective quality evaluation of H.264/AVC FRExt for HD movie content, JVT document JVT—L033, July 2004.
[28] UBVideo website http://www.ubvideo.com.
[29] LSI Logic website: http://www.lsilogic.com.
[30] Microsoft website: http://www.microsoft.com.
[31] Envivio website: http://www.envivio.com.
[32] PixelTools Corporation website: http://www.pixeltools.com.
[33] Nagravision website: http://www.nagravision.com.
[34] Philips website: http://www.philips.com.
[35] Polycom website: http://www.polycom.com.
[36] MainConcept website: http://www.mainconcept.com.
[37] Amphion website: http://www.amphion.com.
[38] Ligos Corporation website: http://www.ligos.com.
[39] LifeSize website: http://www.lifesize.com.
[40] Broadcom website: http://www.broadcom.com.
[41] Netvideo website: http://www.netvideo.com.
[42] Motorola website: http://www.motorola.com.

[43] Polycom website: http://www.polycom.com
[44] Impact Labs Inc. website: http://www.impactlabs.com.
[45] Vanguard Software Solutions website: http://www.vsofts.com.
[46] STMicroelectronics website: http://us.st.com.
[47] M. Dietz, S. Meltzer, CT-aacPlus—a state-of-the-art Audio coding scheme, EBU Technical Review, July 2002, http://www.ebu.ch/trev_291-dietz.pdf.
[48] XM Satellite Radio, http://www.xmradio.com.
[49] S. Meltzer, R. Böhm, F. Henn, SBR enhanced audio codecs for digital broadcasting such as Digital Radio Mondiale (DRM), in 112th AES Convention, Munich, May 2002.
[50] ETSI TS 101 980 v1.1.1 (2001–2009), Digital Radio Mondiale (DRM); System Specification, ETSI, 2001.
[51] T. Ziegler, A. Ehret, P. Ekstrand, M. Lutzky, Enhancing mp3 with SBR: Features and Capabilities of the new mp3PRO Algorithm, in 112th AES Convention, Munich, May 2002.