AN OVERVIEW OF H.264 / MPEG-4 PART 10

(Keynote Speech)

A. Tamhankar and K. R. Rao

University of Texas at Arlington
Electrical Engineering Department
Box 19016, Arlington, Texas 76019, USA
arundhati@ieee.org, rao@uta.edu

Abstract: *The video coding standards to date have not been able to address all the needs required by varying bit rates of different applications and the at the same time meeting the quality requirements. An emerging video coding standard named H.264 or MPEG-4 part 10 aims at coding video sequences at approximately half the bit rate compared to MPEG-2 at the same quality. It also aims at having significant improvements in coding efficiency, error robustness and network friendliness. It makes use of better prediction methods for Intra (I), Predictive (P) and Bi-predictive (B) frames. All these features along with others such as CABAC (Context Based Adaptive Binary Arithmetic Coding) have resulted in having a 2:1 coding gain over MPEG-2 at the cost of increased complexity.*

1. INTRODUCTION

Two international organizations have been heavily involved in the standardization of image, audio and video coding methodologies namely - ISO/IEC and ITU-T. ITU-T Video Coding Experts Group (VCEG) develops international standards for advanced moving image coding methods appropriate for conversational and non-conversational audio/video applications. It caters essentially to real time video applications. ISO/IEC Moving Picture Experts Group (MPEG) develops international standards for compression and coding, decompression, processing, representation of moving pictures, images, audio and their combinations [5]. It caters essentially to video storage, broadcast video, video streaming (video over internet/DSL/wireless) applications. ITU-T has been working on a video coding standard called H.26L since 1997. In August 1998, the first test model was ready and was demonstrated at MPEG's open call for technology in July 2001. In late 2001, ISO/IEC MPEG and ITU-T VCEG decided on a joint venture towards enhancing standard video coding performance – specifically in the areas where bandwidth and/or storage capacity are limited. This joint team of both the standard organizations is called Joint Video Team (JVT). The standard thus formed is called H.264/MPEG-4 part 10 and is presently referred to as JVT/H.26L/Advanced Video Coding (AVC).

The JVT has had several meetings till now- at Fairfax, Virginia in May 2002, Klagenfurt, Austria in July 2002, Geneva, Switzerland in October 2002, Awaji Island, Japan in December 2002 and Pattaya, Thailand in March 2003. The meeting in Japan in December 2002 marks the completion of the design and draft for the standard. Since the meeting in May 2002, the technical specifications are almost frozen and by May 2003 both the organizations will have given their final approval. Table 1 shows the image and video coding standards along with the year they were standardized and their major applications.

Table 1. List of image and video coding standards

Standard	Main Applications	Year
JPEG	Image	1992
JPEG Extensions	Image	1996
JPEG LS part I	Image	1998
part II		1999
JBIG I	Fax	1995
MRC (Mixed Raster Content)	Color Fax, Internet Fax	1998
JBIG2	Fax	2000
H.261	Video Conferencing	1990
H. 262	DTV, SDTV	1995
(MPEG-2)		
H. 263 (Baseline)	Videophone	1998
H. 262+ (Profile 3)		1999
H. 263++ (Profile 5)		2000
H. 26L	VLBR video	2002
MPEG-1	Video CD	1992
MPEG-2	(Generic)	1995
	DTV, SDTV,	
	HDTV, DVD	
MPEG-4 version 2	Interactive video	1999
	(synthetic and natural)	2000
MPEG-7	Multimedia content description interface	2001
MPEG-21	Multimedia Framework	2002
H.264/MPEG4-Part 10	Advanced video coding	2003

H.264/MPEG-4 part 10 applications for video content include but are not limited to the following [21]:

CATV	Cable TV on optical networks, copper, etc.
DBS	Direct broadcast satellite video services
DSL	Digital subscriber line video services
DTTB	Digital terrestrial television broadcasting
ISM	Interactive storage media (optical disks, etc.)
MMM	Multimedia mailing
MSPN	Multimedia services over packet networks
RTC	Real-time conversational services (videoconferencing, videophone, etc.)
RVS	Remote video surveillance
SSM	Serial storage media (digital VTR, etc.)

The main goals of JVT are: significant coding efficiency, simple syntax specifications and seamless integration of video coding into all current protocols and multiplex architectures (network friendliness). This paper is intended to serve as a tutorial for H.264/MPEG-4 part 10. The paper will consist of five sections. In section 2, the main requirements from a video coding standard and H.264/MPEG-4 part 10 features meeting these requirements are listed, section 3 concentrates on the basic architecture of H.264/MPEG-4 part 10, section 4 delves into the main concepts involved in H.264/MPEG-4 part 10. Section 5 presents comparison of

H.264/MPEG-4 part 10 with existing standards and the results from experiments and analysis conducted thus far.

2. REQUIREMENTS AND FEATURES OF H.264 / MPEG-4 PART 10

Requirements for H.264/MPEG-4 part 10 arise from the various video applications that it aims at supporting like video streaming, video conferencing, over fixed and wireless networks and over different transport protocols. H.264/MPEG-4 part 10 features thus aim at meeting the requirements evolving from such applications. The following lists the important requirements and how H.264/MPEG-4 part 10 meets them.

2.1 Robust (Error Resilient) Video Transmission using Parameter set

One of the key problems faced by previous standards is their layered nature, which results in less robust video transmission in packet lossy environments [3]. Previous standards contained header information about slice/picture/GOP/sequence that was coded at the start of each slice/picture/GOP/sequence. The loss of packet containing this header information would make the data dependent on this header, as useless. H.264/MPEG-4 part 10 overcame this shortcoming by making the packets transmitted synchronously in a real-time multimedia environment as self-contained. That is, each packet can be reconstructed without depending on the information from other packets. All information at higher layers is system-dependent, but not content-dependent and is conveyed asynchronously. Parameters that change very frequently are added to the slice layer. All other parameters are collected in a "Parameter Set". H.264/MPEG-4 part 10 standard specifies a method to convey Parameter Sets in a special Network Abstraction Layer (NAL) unit type. NAL is described in the next section. Different logical channels or out-of-band control protocols may be used to convey parameter sets from the coder to the decoder. In-band parameter set information and out-of-band control protocol should not be used in combination.

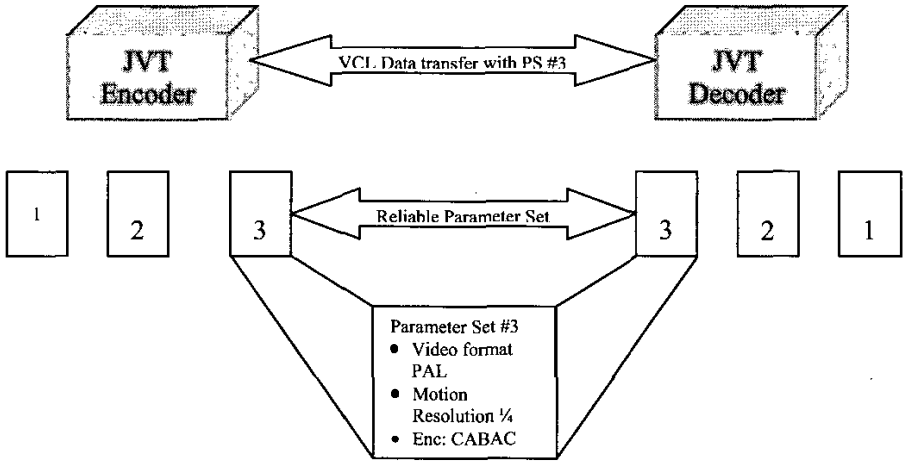


Fig. 1. Parameter Set Concept [3]

2.2 Network friendliness

Previous video coding standards, such as H.261, MPEG-1, MPEG-2 and H.263 [6] were mainly designed for special applications and transport protocols usually in a circuit-switched, bit-stream oriented environment. JVT experts realized the growing importance of packet-based data over fixed and wireless networks right in the beginning and designed the video codec from that perspective. Common test cases for such transmission include fixed Internet conversational services as well as packet-switched conversational services and packet-switched streaming services over 3G mobile networks. These IP-networks usually employ IP on the network layer, UDP at the transport layer and RTP at the application layer. IP and UDP offer an unreliable datagram service while RTP makes the transport of media possible. Sequence numbers are used to restore the out-of-order IP packets. RTP payload does not add to the bit stream but specifies how the media information should be interpreted. The standardization process of both JVT codec and RTP payload specifications for H.264/MPEG-4 part 10 is still an ongoing issue but the goal of designing a simple coding scheme should be achieved. [3]

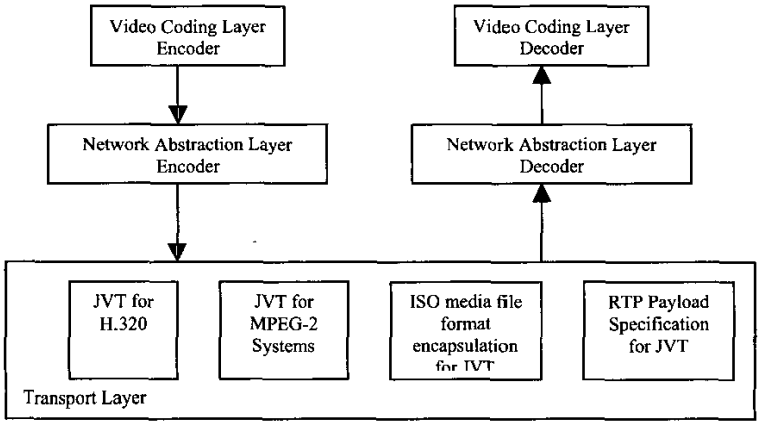


Fig. 2. JVT Coding in Network Environment [3]

2.3 Support for different bit rates, buffer sizes and start-up delays of the buffer

In many video applications, the peak bit rate varies according to the network path and also fluctuates in time according to network conditions. In addition, the video bit streams are delivered to a variety of devices with different buffer capabilities. A flexible decoder buffer model, as suggested for H.264/MPEG-4 part 10 in [4] would support this wide variety of video application conditions – bit rates, buffer sizes and start-up delays of the buffer. H.264/MPEG-4 part 10 Video coding standard requires that the bit stream to be transmitted should be decodable by a Hypothetical Reference Decoder (HRD) without an underflow or overflow of the reference buffer. This aspect of the decoder is analyzed in [4] in order to support different bit rates.

The HRD Coded Picture Buffer represents a means to communicate how the bit rate is controlled in the process of compression. The HRD contains Coded Picture Buffers (CPB) through which compressed data flows with a precisely specified arrival and removal timing,

as shown in Fig. 4 (HRD Buffer Verifiers). An HRD may be designed for variable or constant bit rate operation, and for low-delay or delay-tolerant behavior.

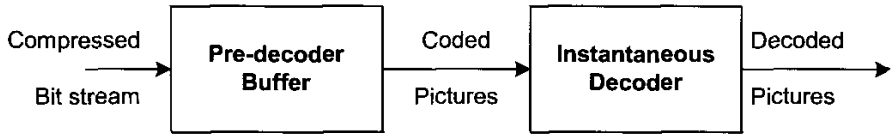


Fig. 3. A Hypothetical Reference Buffer [1]

The HRD contains the following buffers, as shown in Fig. 4.

- One or more Coded Picture Buffers (CPB), each of which is either variable bit rate (VBR) or constant bit rate (CBR), and
- One Decoded Picture Buffer (DPB) attached to the output of one of the CPBs.

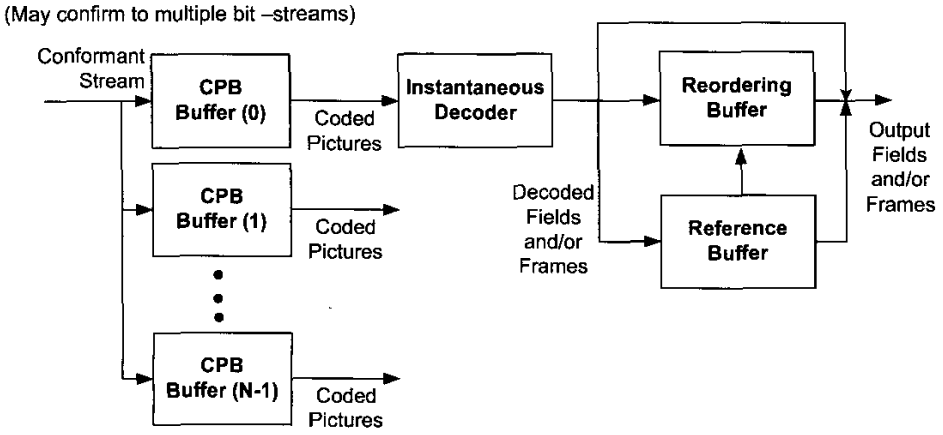


Fig. 4. HRD Buffer Verifiers [1]

The multiple CPBs exist because a bit stream may conform to multiple CPBs. Considering the operation of a single CPB, data associated with coded pictures flow into the CPB according to a specified arrival schedule. Each coded picture is removed instantaneously and decoded by the decoder at CPB removal times. Decoded pictures are placed in the DPB at the CPB removal time. Finally, pictures are removed from the DPB towards the end of the DPB output time and the time that they are no longer needed as reference for decoding. The primary conditions are that the CPB neither underflow nor overflow and that the DPB does not overflow.

2.4 Improved Prediction

Earlier standards used maximum two prediction signals, one from a past frame and the other from a future frame in the bi-directional mode (B – picture) as shown in Fig. 5 [14]. H.264/MPEG-4 part 10 allows multiple reference frames for prediction. Maximum of five reference frames may be used for prediction. Although this increases the complexity of the encoder, the encoder remains simple and the prediction is significantly improved. A multiple reference case of prediction signals from two past frames is illustrated in Fig. 6 [14]. Figure 7

[14] shows the graph for luminance PSNR vs bit rate for the CIF video sequence Mobile and Calendar (30fps) compressed with H.264/MPEG-4 part 10. The graph shows that the multiple reference prediction always gives a better performance – improvement of around 0.5 dB for low bit rates of the order of 0.5Mbps and around 1dB for higher bit rates of the order of 2.5Mbps.

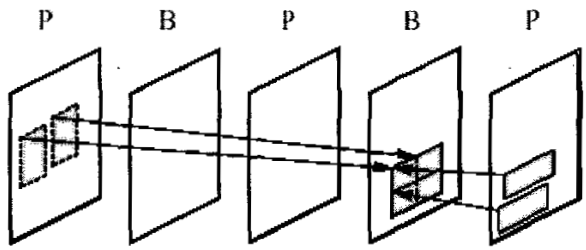


Fig. 5. A bi-directional mode allows a linear combination of one past and one future macroblock prediction signal [14]

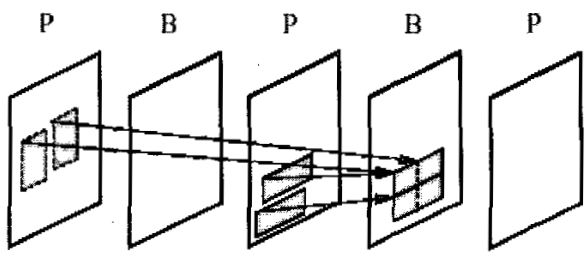


Fig. 6. Multiple reference mode also allows a linear combination of two past macroblock prediction signals [14]

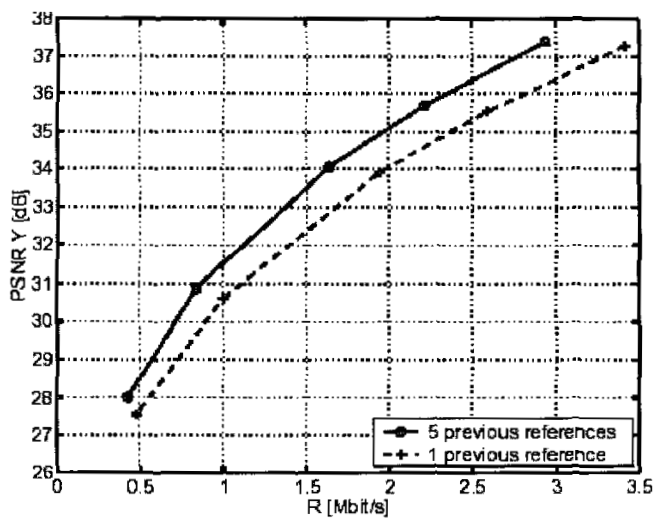


Fig. 7. Luminance PSNR vs bit rate for the CIF video sequence Mobile and Calendar (30fps) compressed with H.264/MPEG-4 part 10 [14]

2.5 Improved Fractional Accuracy

Fractional pel values add significantly to the accuracy of the reconstructed image. These are imaginary pel positions assumed to be stationed between physical pels. Their values are evaluated using interpolation filters. Previous standards have incorporated half-pel and quarter-pel accuracies. H.264/MPEG-4 part 10 improves prediction capability by incorporating quarter-pel accuracies. This would increase the coding efficiency at high bit rates and high video resolution. [1]

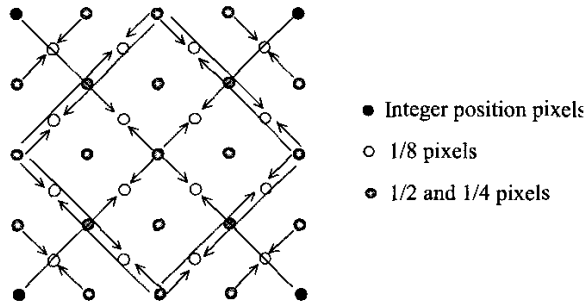


Fig. 8. Integer and fractional position of pixels in a block [1]

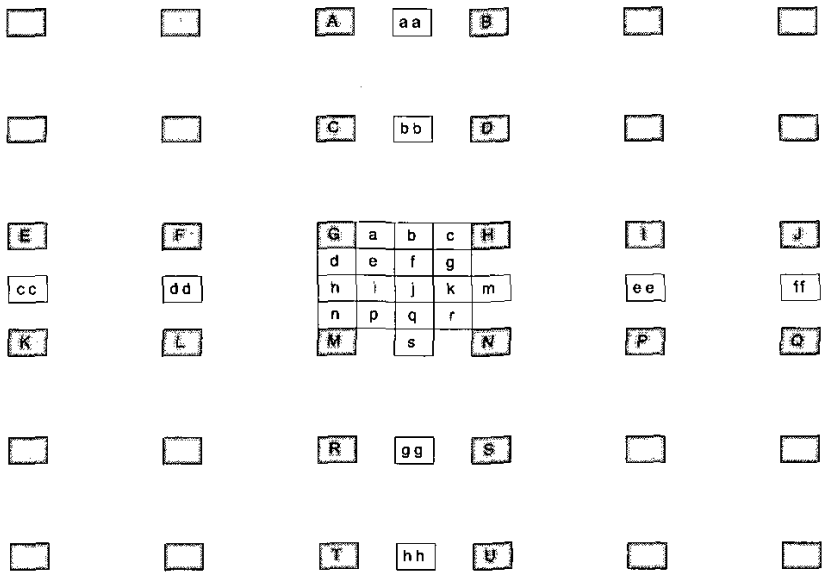


Fig. 9. Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter sample luma interpolation. Half-sample pels include b, h, m. Quarter-sample pels include a, c, d, n [21]

2.6 Significant data compression

Earlier standards implemented quantization steps with constant increments. H.264/MPEG-4 part 10 includes scalar quantizer with step sizes that increase at the rate of 12.5%.

Chrominance values have finer quantizer steps. This provides significant data compression. [1]

2.7 Better Coding Efficiency

H.264/MPEG-4 part 10 uses UVLC (Universal Variable Length Coding), CAVLC (Context-based Variable Length Coding) and CABAC (Context based Adaptive Binary Arithmetic Coding) to provide efficient entropy coding. CABAC will be discussed in detail in section 4. CABAC provides as good as 2:1 coding gain over MPEG-2.

2.8 Overlay Coding Technique

Faded transitions are such that the pictures of two scenes are laid on top of each other in semi-transparent manner, and the transparency of the pictures at the top gradually changes in the transition period. Motion compensation is not a powerful enough method to represent the changes between pictures in the transition during a faded scene. H.264/MPEG-4 part 10 utilizes overlay coding technique that provides over 50% bit-rate savings in both cross-fades and through-fades compared to earlier techniques [9]. Figure 10 [9] shows that overlay coding outperforms weighted B-picture averaging remarkably. Over 50% bit-rate savings in the scene transition period can be achieved. BT-PAW overlay coding performs better than B-PAW overlay coding that outperforms simple overlay coding. Figure 11 [9] shows the 12th picture of the Foreman-to-Carphone cross-fade to show that the subjective quality can be greatly improved at a similar bit-rate by using BT-PAW overlay coding.

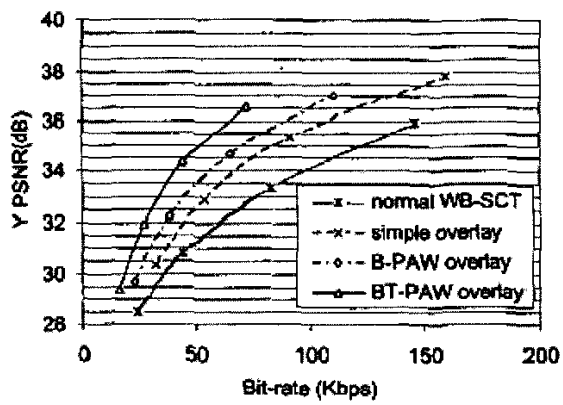


Fig. 10. Cross-fade from Foreman to Carphone for different overlays [9]

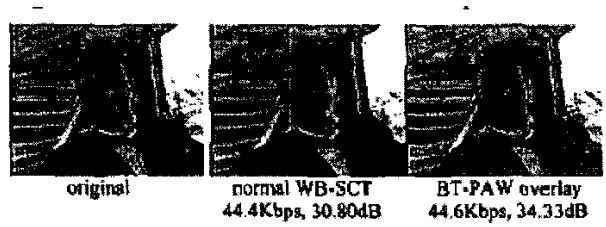


Fig. 11. The 12th picture of the Foreman-to-Carphone cross-fade [9]

2.9 Better Video Quality

H.264/MPEG-4 part 10 employs blocks of different sizes and shapes, higher resolution fractional-pel motion estimation and multiple reference frame selection. These provide better motion estimation/compensation. On the transform front, H.264/MPEG-4 part 10 uses integer based transform, which approximates the DCT used in other standards besides eliminating the mismatch problem in its inverse transform. Hence the video received is of very good quality compared to the previous standards.

Figures 12 and 13 [12] show BMP (Bit Map Picture) images of the decoded pictures for two cases. These two images are encoded at the same bit rate (about 400 kbps). PSNR of Fig.12 is 29.25dB and Fig.13 28.69dB. PSNR of Fig.12 is about 0.5dB higher than that of Fig.13. Comparing the visual quality, the subjective quality of Fig.12 is better. The subjective quality is improved at high frequency and the edge of circle at low frequency. At left bottom and top right on Fig. 12, some artifacts can be seen. However in Fig. 13, the artifact is not visible at the same parts.

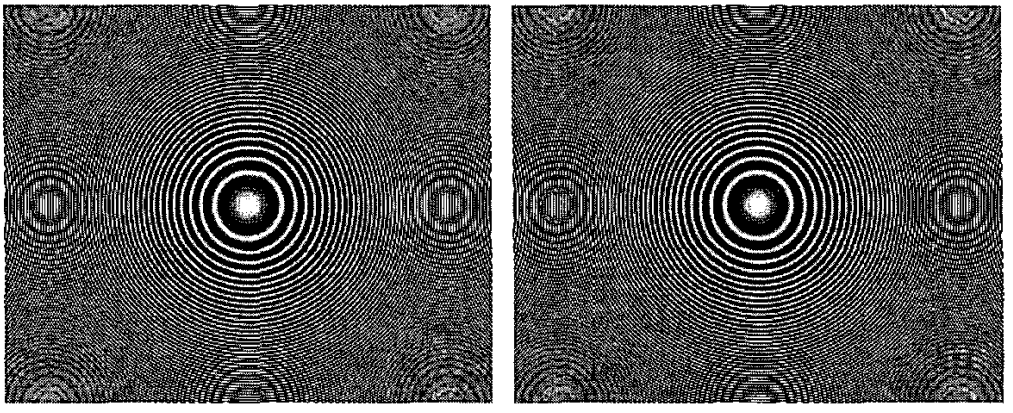


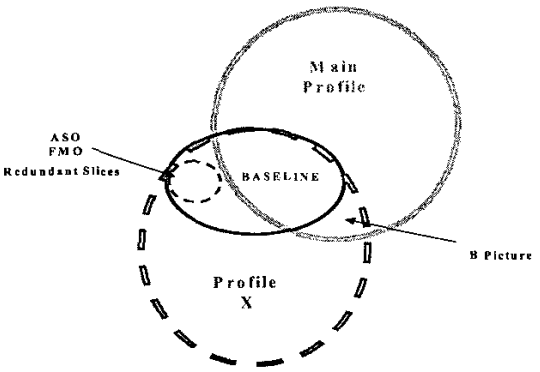
Fig. 12. Decoded picture of TML+WM (QP=16) Fig. 13. Decoded picture of TML (QP= 20) [12]

2.10 Group capabilities

H.264/MPEG-4 part 10 grouped its capabilities into profiles and levels – Baseline, Main and Extended profile [21]. A profile is a subset of the entire bitstream of syntax that is specified by the International Standard. Within each profile there are a number of levels designed for a wide range of applications, bit rates, resolutions, qualities and services. A "level" has a specified set of constraints imposed on parameters in a bitstream. It is easier to design a decoder if the profile, level and hence the capabilities are known in advance. The three profiles and their proposed area of application are shown in Table 2.

Table 2. H.264/MPEG-4 part 10 profiles and their major application areas

Profile	Applications
Baseline	Video Conferencing, Video Telephony
Main	Broadcast Video
Extended	Streaming Media



X = Extended profile

Fig. 14. Overlapping features of H.264/MPEG-4 part 10 profiles

The baseline profile is intended to be the common profile supported by all implementations. Reference [15] suggests that the "baseline profile" and its levels would be royalty-free for all implementations. Section 4.1 deals more in detail with the profiles used with H.264/MPEG-4 part 10.

3. BASIC ARCHITECTURE OF THE STANDARD

Conceptually, H.264/MPEG-4 part 10 consists of two layers – Video Coding Layer (VCL) and Network Abstraction Layer (NAL). VCL is the core coding layer, which concentrates on attaining maximum coding efficiency. NAL abstracts the VCL data in terms of the details required by the transport layer and to carry this data over a variety of networks. The VCL layer takes care of the coding of transform coefficients and motion estimation/compensation information. NAL provides the header information about the VCL format, in a manner that is appropriate for conveyance by the transport layers or storage media. A NAL unit (NALU) defines a generic format for use in both packet-based and bit-streaming systems. The format for both the systems is the same except that the NAL unit in a bit-stream system can be preceded by a start code.

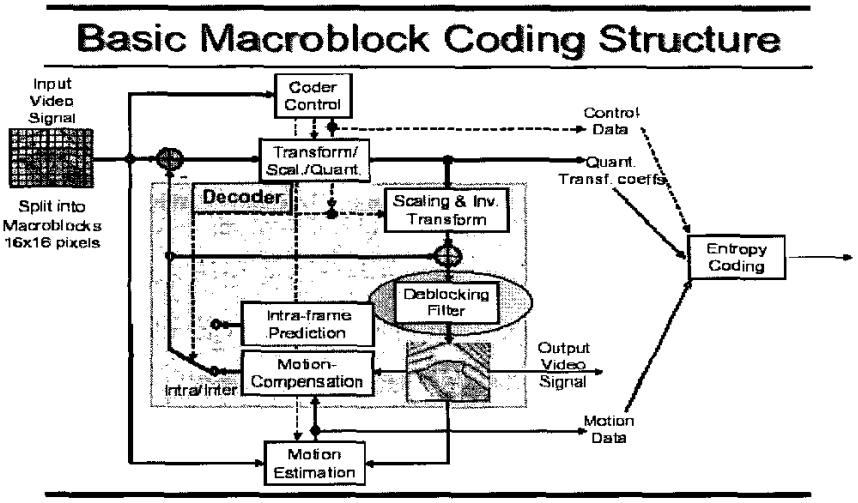


Fig. 15. Basic Block Diagram of H.264/MPEG-4 part 10 Encoder [10]

The basic block diagram of a H.264/MPEG-4 part 10 coder is shown in Fig. 15. Common video formats used are CIF and QCIF as shown in Figs. 16 and 17. The luminance and chrominance blocks follow the 4:2:0 format. A picture is divided into macroblocks of 16x16 luma samples with two associated 8x8 chroma samples. A set of macroblocks forms a slice. Each macroblock belongs to exactly one slice. The minimum number of slices for a picture is one.

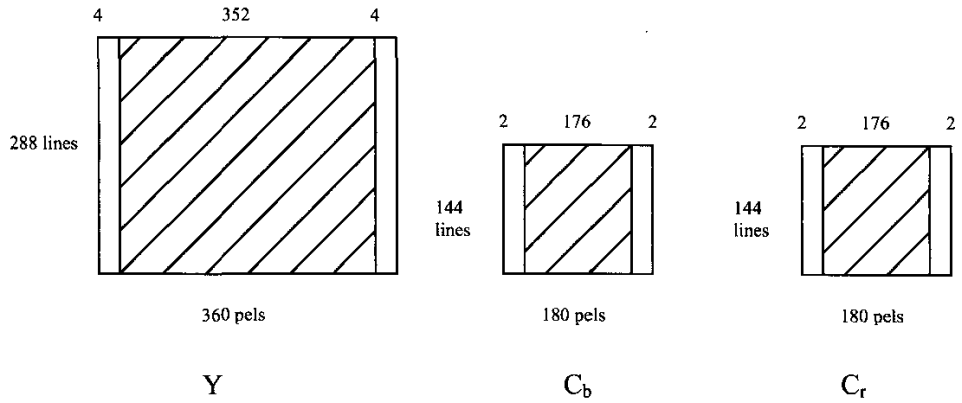


Fig. 16. CIF (Common Intermediate Format)

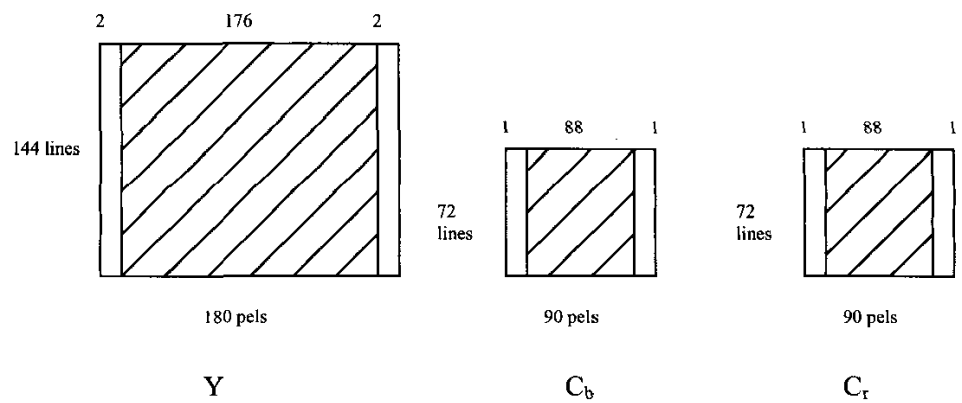


Fig. 17. QCIF (Quadrature Common Intermediate Format)

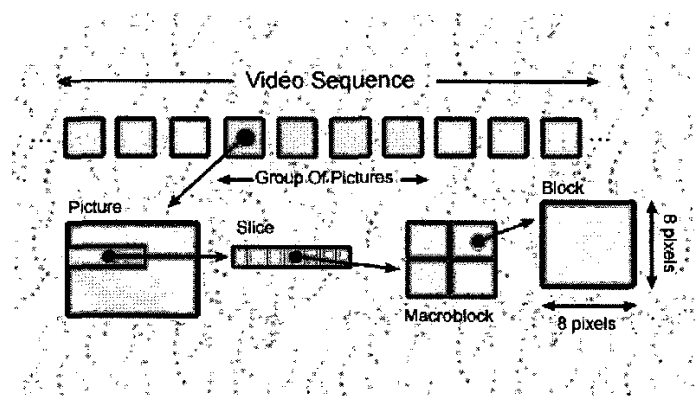


Fig. 18. Sequence-GOP-picture-slice-macroblock-block

Each macroblock (16x16) can be partitioned into blocks with sizes 16x16, 16x8, 8x16 and 8x8. An 8x8 block can further be sub-partitioned into sub-blocks with sizes 8x8, 8x4, 4x8 and 4x4 as shown in Fig. 19. Each block is motion compensated using a separate motion vector. The numbering of the motion vectors for the different blocks depends on the inter mode. For each block, the horizontal component comes first followed by the vertical component.

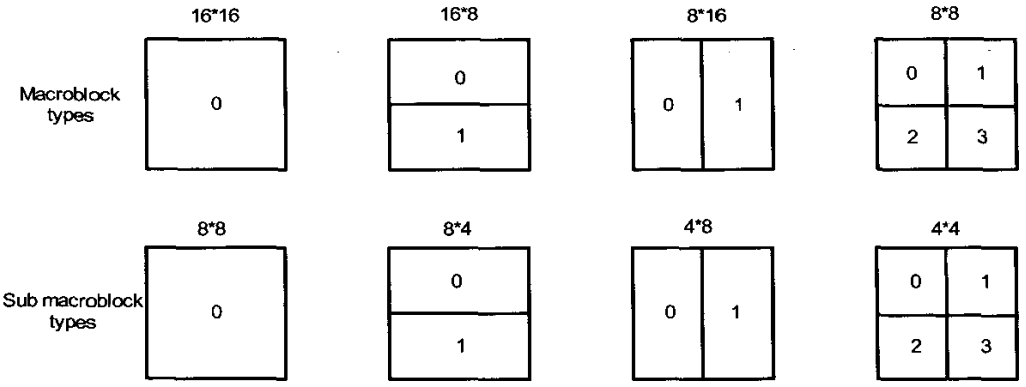


Fig. 19. Macroblock sub division type and block order [1]

A coded block pattern (CBP) contains information of which 8x8 blocks – luminance and chrominance – contain transform coefficients. Notice that an 8x8 block contains four 4x4 blocks meaning that the statement '8x8 block contains coefficients' means that 'one or more of the four 4x4 blocks contain coefficients'. The four least significant bits of CBP contain information on which of the four 8x8 luminance blocks in a macroblock contain nonzero coefficients. A 0 in position n of CBP means that the corresponding 8x8 block has no coefficients whereas a 1 means that the 8x8 block has at least one non-zero coefficient.
For chrominance, 3 possibilities are defined in [16] as:
nc = 0 : No chrominance coefficients at all
nc = 1 : There are nonzero 2x2 transform coefficients. All chroma AC coefficients = 0. No EOB for chrominance AC coefficients is sent
nc = 2 : There may be 2x2 nonzero coefficients. At least one AC coefficient is non-zero. 10 End of Blocks (EOB) (2 for DC coefficients and 2x4 = 8 for the eight 4x4 blocks)
Statistics of CBP values in case of Intra and Inter are different and hence different codewords are used.

Data partitioning may or may not be used with a slice. When data partitioning is not used, the coded slices start with a slice header and are followed by the entropy-coded symbols of the macroblock data for all the macroblocks of the slice in raster scan order. When data partitioning is used, the macroblock data of a slice is partitioned in up to three partitions – header information; intra coded block pattern (CBP) and coefficients; and inter coded block pattern and coefficients. The order of the macroblocks in the transmitted bit stream depends on the macroblock allocation map (MAP). The macroblock allocation map consists of an array of numbers one for each coded macroblock indicating the slice group to which the coded macroblock belongs [1].

The H.264/MPEG-4 part 10 coder involves motion estimation and compensation in order to best provide a prediction for a block. The transform coefficients of a block are scaled and

quantized and later entropy coded for transmission. Before entropy coding, the transform coefficients are reconstructed. A de-blocking filter is applied to all reconstructed macroblocks of a picture and then stored for reference. These stored frames may then be used later for motion estimation and compensation. Reference [32] indicates that 52 QP (Quantization Parameter) values can be used, from –12 to +39. There is increase in step size of about 12% from one QP to the next and there is no "dead zone" in the quantization process. The quantizer value can be changed at the macroblock level. Different QP values are used for luma and chroma.

Coding a picture involves the decision as to whether it is to be Inter or Intra coded; coded with or without motion compensation. Intra-predictive picture coding indicates entropy coding of the transform coefficients of the blocks, without prediction from any reference frame, in estimating the coefficient values. Inter-predictive picture coding indicates using reference frames stored in the buffer to predict the values of the current frame. Motion compensation may be used when the motion compensated macroblock follows the mean square error (MSE) limit given by

$$\mathrm{MSE} = \frac{1}{256}\sum_{m=0}^{15}\sum_{n=0}^{15}\left(x_0(m,n) - x_{mc}(m,n)\right)^2$$

where $x_0(m,n)$ denotes original macroblock and $x_{mc}(m,n)$ denotes the motion compensated macroblock. The mean square error along with the variance of the input (VAR input) macroblock may be used in deciding whether Intra or Inter mode is used. Intra mode is used when the input variance is smaller and the MSE is greater. Please refer to [6] for further details. Each block is motion compensated using a separate motion vector.

Intra Prediction

It is referred to as an Intra (I) picture. It uses only transform coding and the neighboring blocks of the same frame to predict block values. Intra prediction is performed on 16x16 luma and 4x4 luma blocks. No intra prediction is performed on 8x8 luma blocks.

Intra prediction for 4x4 luma block

A 4x4 luma block contains samples as shown in Fig. 20. There are nine intra prediction modes as shown in Table 3. Modes 0, 1, 3, 4, 5, 6, 7, and 8 are directional prediction modes as indicated in Fig. 21. Mode 2 is 'DC-prediction'.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 20. Identification of samples used for intra spatial prediction [1]

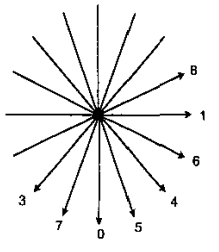


Fig. 21. Intra-prediction directions [1]

Table 3. Intra prediction modes for 4x4 luma block

Mode Number	Mode Name
0	Vertical
1	Horizontal
2	DC
3	Diagonal down/left
4	Diagonal down/right
5	Vertical –right
6	Horizontal-down
7	Vertical-left
8	Horizontal- up

To illustrate Intra prediction, let us consider mode 0: vertical prediction

This mode shall be used only if A, B, C, D are available. The prediction in this mode shall be as follows:

- a, e, i, m are predicted by A,
- b, f, j, n are predicted by B,
- c, g, k, o are predicted by C,
- d, h, l, p are predicted by D.

The mode 2: DC prediction has the following rules:

If all samples A, B, C, D, I, J, K, L, are available, all samples are predicted by (A+B+C+D+I+J+K+L+4)>>3. If A, B, C, and D are not available and I, J, K, and L are available, all samples shall be predicted by (I+J+K+L+2)>>2. If I, J, K, and L are not available and A, B, C, and D are available, all samples shall be predicted by (A+B+C+D+2)>>2. If all eight samples are not available, the prediction for all luma samples in the 4x4 block shall be 128. A block may therefore always be predicted in this mode.

Please refer to reference [1] for more details about each mode.

Intra prediction for 16x16 luma block

One of the four prediction modes listed in Table 4 are applied to a luma block. Please refer to [1] for more details about each mode.

Table 4. Intra prediction modes for 16x16 luma block

Mode number	Mode name
0	Vertical
1	Horizontal
2	DC
3	Plane

Input to the prediction process are samples constructed prior to the deblocking process from neighbouring luma blocks (if available). There are 33 neighboring samples for a 16x16 macroblock with x = -1, y = -1..15 and with x = 0..15, y = -1. Outputs of this process are Intra

prediction luma samples for the current macroblock $pred_L[x, y]$. To illustrate the prediction modes, let us consider the vertical prediction mode.

The Vertical mode shall be used only when the samples p[x, -1] with x = 0..15 are marked as "available for Intra_16x16 prediction". The prediction will be as follows:

$pred_L[x, y] = p[x, -1]$, with x, y = 0..15

Please refer to reference [40] for more details about each mode.

Intra prediction for Chroma

Intra prediction for chroma blocks support only one mode. A 8x8 chroma macroblock consists of four 4x4 blocks A, B, C, D as shown in the figure 22 below. S0, S1, S2 and S3 are the sum of four neighboring samples.

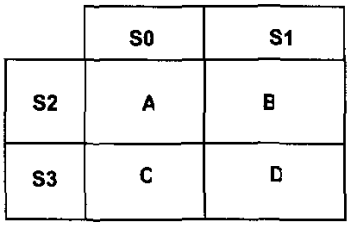


Fig. 22. Intra Prediction for Chroma macroblock [40]

There are four prediction cases depending upon whether S0, S1, S2 or S3 are inside or outside. For example, if all are inside:

A = (S0 + S2 + 4)/8
B = (S1+ 2)/4
C = (S3 + 2)/4
D = (S1 + S3 + 4)/8

Inter-prediction

Inter-prediction may be performed on 16x16, 16x8, 8x16, 8x8 blocks and 4x8, 8x4 and 4x4 sub blocks. There are different types of inter-prediction pictures depending on the reference pictures used in the prediction.

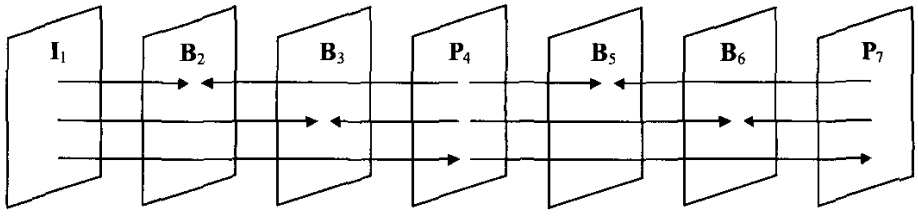


Fig. 23. P, B picture prediction concept [1]

P-picture

As in previous standards, a Prediction (P) picture uses a previous encoded P or I picture for prediction. H.264/MPEG-4 part 10 allows single or multiple reference frames, a maximum of five reference frames. In case of multiple reference frames, interpolation filters are used to predict the transform coefficients. It is coded using forward motion compensation. To encode a macroblock in a P picture, the set of possible macroblock types are given as:

S_{MB} = { SKIP, INTER_16x16, INTER_16x8, INTER 8x16, INTER_8x8, INTER_8x4, INTER_4x8, INTER_4x4, INTRA_4x4, INTRA_16x16 }

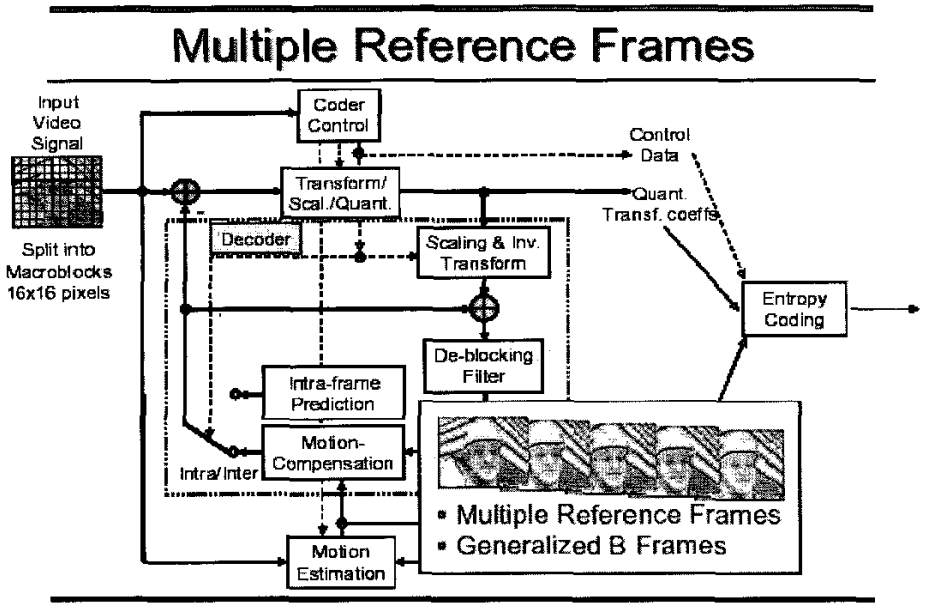


Fig. 24. Block Diagram illustrating multiple reference frames [10]

B-picture

As in previous standards, a B picture uses both past and previous pictures as reference. Thus it is called Bi-directional prediction. It uses both forward and backward motion compensation. However unlike previous standards, a B picture can utilize B, P or I picture for prediction. There are five prediction types supported by B-picture. They are forward, backward, bi-predictive, direct and intra prediction modes. Forward prediction indicates that the prediction signal is formed from a reference picture in the forward reference frame buffer. A picture from backward reference frame buffer is used for backward prediction. In both direct and bi-predictive modes, prediction signal is formed by weighted average of a forward and backward prediction signal. The only difference is that the bi-predictive mode has separate encoded reference frame parameters and motion vectors for forward and backward, whereas in the direct mode, the reference frame parameters, forward and backward motion vectors for the prediction signals are derived from motion vectors used in the macroblock of the picture. [1]

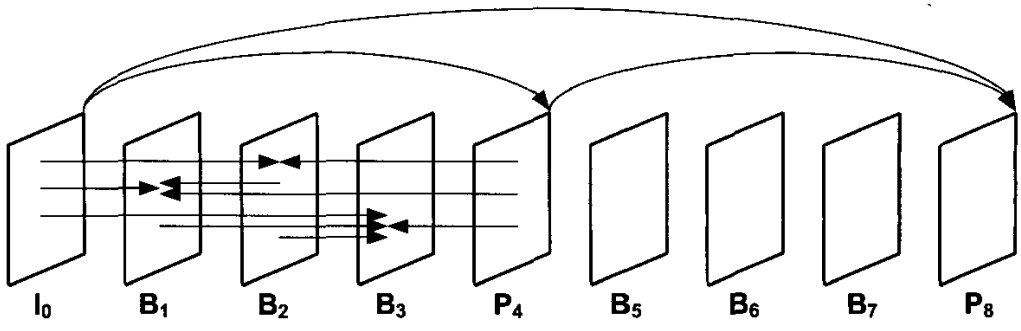


Fig. 25. Illustration of B picture prediction

Note: The order of displaying these pictures is different from the order in which they are coded. An I picture is coded first. Then the P picture dependent on this I picture's prediction signal, is coded next. Lastly the B picture is predicted from the already coded P and I pictures.

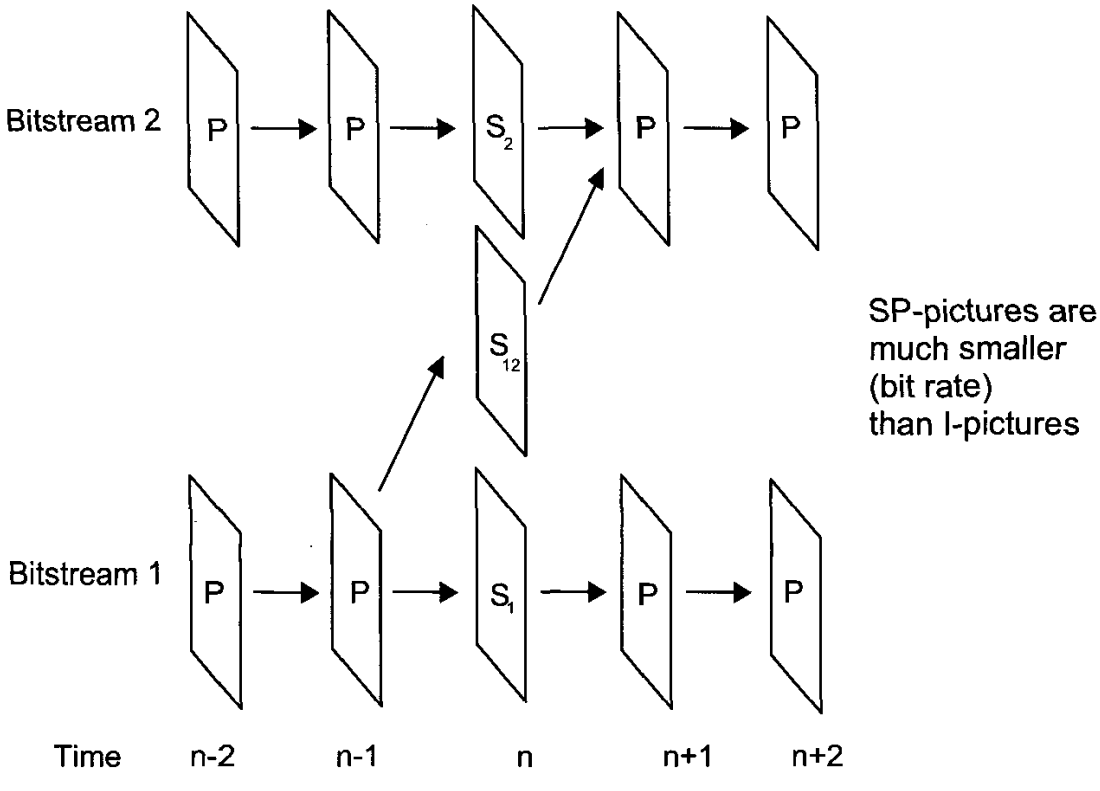


Fig. 26. S picture [1]

S-picture

It is a new picture introduced for H.264/MPEG-4 part 10. It is used in the Extended profile. Switched (S) pictures are of two types: Switched I (SI) – picture and Switched P (SP) - picture. SP coding makes use of temporal redundancy through motion-compensated inter prediction from previously-decoded reference pictures, using at most one motion vector and reference picture index to predict the sample values of each block, encoded such that it can be reconstructed identically to another SP slice or SI slice. SI slice makes use of spatial prediction and can identically reconstruct to another SP slice or SI slice. The inclusion of these pictures in a bit-stream enables efficient switching between bit streams with similar content encoding at different bit rates, as well as random access and fast play back modes. Macroblock modes for SP-picture are similar to P-picture whereas SI-picture has 26 macroblock modes shown in Table 5.

Table 5. MB –type for SI pictures [17]

Code_number	mb_mode (SI-pictures)
0	SIntra 4x4
1	Intra 4x4
2	0,0,0
3	1,0,0
4	2,0,0
5	3,0,0
6	0,1,0
7	1,1,0
8	2,1,0
9	3,1,0
10	0,2,0
11	1,2,0
12	2,2,0
13	3,2,0
14	0,0,1
15	1,0,1
16	2,0,1
17	3,0,1
18	0,1,1
19	1,1,1
20	2,1,1
21	3,1,1
22	0,2,1
23	1,2,1
24	2,2,1
25	3,2,1

The following are the macroblock modes for SI pictures for 16x16 intra coding. Here SIntra 4x4 stands for SI mode. Intra 4x4 stands for 4x4 Intra coding. The numbers in the parenthesis represent (Imode, AC, nc) as in [17].

Deblocking filter

A conditional filtering is applied to all block edges of a slice, except edges at the boundary of the picture and any edges for which the deblocking filter is disabled as specified by disable_deblocking_filter_idc. Deblocking filter is a mandatory part of the standard. This filtering is done on a macroblock basis, with macroblocks being processed in raster-scan order throughout the picture. For luma, as the first step, the 16 samples of the 4 vertical edges of the 4x4 raster are filtered beginning with the left edge, as shown on the left-hand side of Fig. 27. Filtering of the 4 horizontal edges (vertical filtering) follows in the same manner, beginning with the top edge, as shown on the right-hand side of Fig. 27. The same ordering applies for chroma filtering, with the exception that 2 edges of 8 samples each are filtered in each direction.

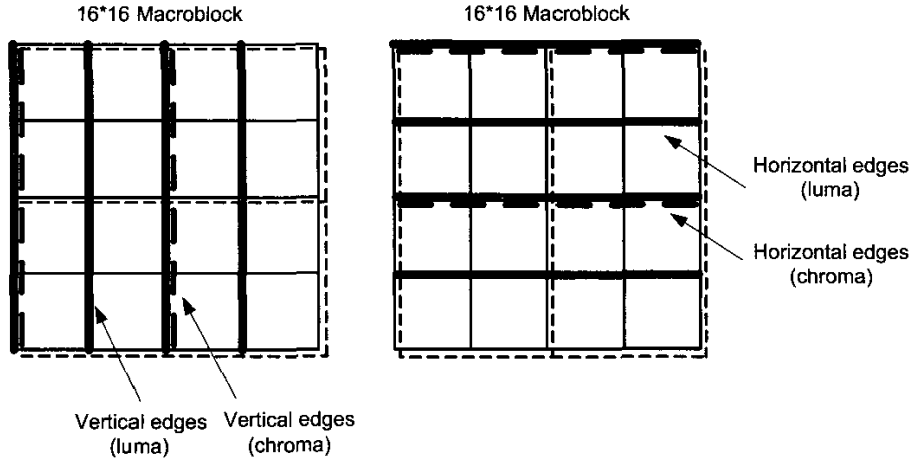


Fig. 27. Boundaries in a macroblock to be filtered (luma boundaries shown with solid lines and chroma boundaries shown with dotted lines) [1]

For each boundary between neighboring 4x4 luma blocks, a "Boundary Strength" Bs is assigned. Every block boundary of a chroma block corresponds to a specific boundary of a luma block. Bs values for chroma are not calculated, but simply copied from the corresponding luma Bs. If Bs=0, filtering is skipped for that particular edge. In all other cases filtering is dependent on the local sample properties and the value of Bs for this particular boundary segment.

For each edge, if any sample of the neighboring macroblocks is coded using Intra macroblock prediction mode, a relatively strong filtering (Bs=3) is applied. A special procedure with even stronger filtering may be applied on macroblock boundaries with both macroblocks coded using Intra macroblock prediction mode (Bs=4). If neither of the neighboring macroblocks are coded using Intra macroblock prediction mode and at least one of them contains non-zero transform coefficient levels, a medium filtering strength (Bs=2) is used. If none of the

previous conditions are satisfied, filtering takes place with Bs=1 if at least one of the following conditions is satisfied:

- one of the neighboring macroblocks is coded in frame mode and the other is coded in field mode
- both macroblocks are coded in frame mode and the prediction of the two blocks is formed using different reference pictures or a different number of reference pictures
- both macroblocks are coded in field mode and the prediction of the same parity field of the two macroblocks is formed using different reference pictures or a different number of reference pictures.
- any corresponding pair of motion vectors from the two neighboring macroblocks is referencing the same picture and either component of this pair has a difference of more than one sample. Otherwise filtering is skipped for that particular edge (Bs=0).

Reference [1] provides further details about deblocking filter.

Reference Frames

Reference frames are stored in the frame buffer. It contains short term and long term frames which may be used for macroblock prediction. With respect to the current frame, the frames before and after the current frame, in the display sequence order are called the "Reverse Reference frame" and "Forward Reference frame" respectively. These frames are also classified as "Short term frame" and "Long term frame".

Memory management is required to take care of marking some stored frames as 'unused' and deciding which frames to delete from the buffer for efficient memory management. There are two types of buffer management modes: Adaptive buffering mode and Sliding window buffering mode. Indices are allotted to frames in the buffer. Each picture type has a default index order for frames. Figure 28 gives an idea of storage of frames in the buffer for *P* picture.

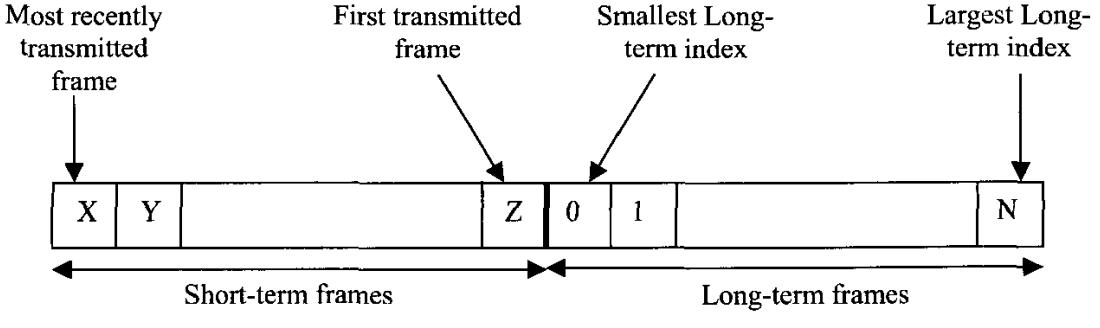


Fig. 28. Example of short-term and long-term frame storage in the buffer

Motion Vector, Motion Estimation, Motion Compensation

Macroblock to be encoded is compared with blocks from previous frames that are stored in the buffer. The process of finding the best match within the search window is called motion estimation. This best match is found from the reference frames stored in the buffer. The selected frames for prediction are indicated by the reference index associated with the frame index in the buffer. The process of encoding, based on the values predicted from the best match is called motion compensation. A motion vector is defined as a vector drawn from the reference. No vector component prediction can take place across macroblock boundaries that do not belong to the same slice. For the purpose of motion vector component prediction, macroblocks that do not belong to the same slice are treated as outside the picture. This motion vector is then encoded and transmitted. Motion compensation is applied to all blocks in a macroblock. Accuracy of quarter-pel and one-eighth-pel resolutions may be used during motion compensation. One-eighth pel motion vector resolution has been dropped in the Geneva meeting (Aug. 2002). Figures 30 and 31 illustrate the motion estimation concept.

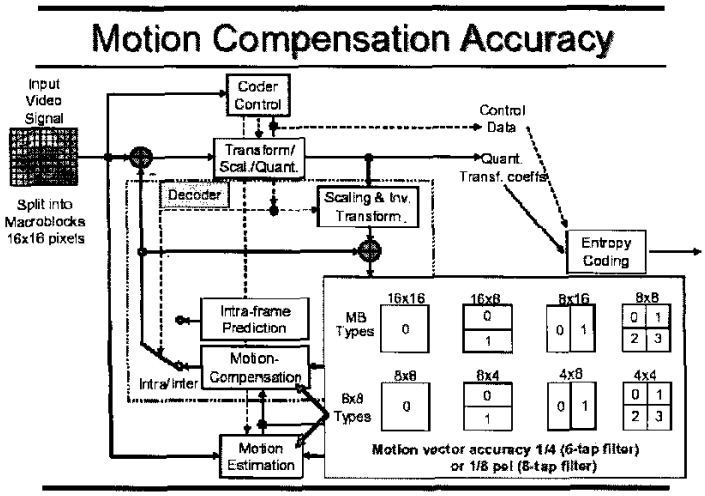


Fig. 29. Block Diagram emphasizing Motion Compensation [10]

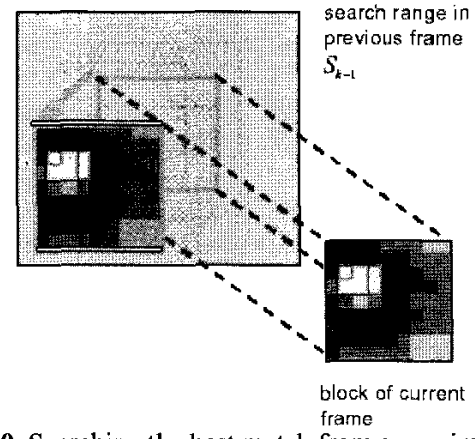


Fig. 30. Searching the best match from a previous frame

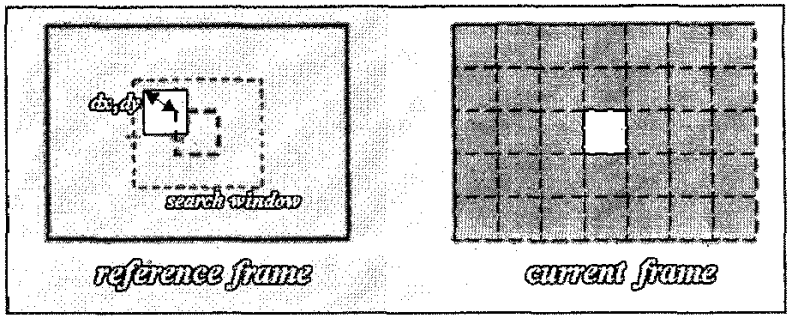


Fig. 31. Block based motion estimation
dx, dy represent the differential motion vectors

Chroma vectors

Chroma motion vectors are derived from the luma motion vectors. Since chroma has half the resolution of luma, the chroma vectors are obtained by dividing the corresponding luma vectors by two. Since the accuracy of luma motion vectors is one-quarter sample and chroma has half resolution compared to luma, the accuracy of chroma motion vectors is one-eighth sample, i.e., a value of 1 for the chroma motion vector refers to a one-eighth sample displacement. [40]

Median and Directional Prediction methods are used for motion vector prediction. Median prediction is used on all blocks except 16x8 and 8x16. Directional prediction is used for 16x8 and 8x16 blocks. Figure 32 indicates motion vectors A, B, C, D and E. These motion vectors can be from different reference pictures. It is required to predict E. The prediction is normally formed as the median of A, B and C.

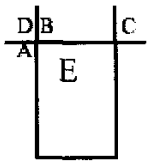


Fig. 32. Median Prediction [21]

The following rules determine the predicted motion vector value resulting from the median prediction process for block E:

- If block C is outside the current picture or slice or is not available due to the decoding order within a macroblock, its motion vector and reference picture index shall be considered equal to the motion vector and reference picture index for block D.
- If blocks B, C, and D are all outside the current picture or slice, their motion vector values and reference picture indices shall be considered as equal to the motion vector value and reference picture index for block A.
- If any predictor not specified by the first or second rules above is coded as intra or is outside the current picture or slice, its motion vector value shall be considered equal to zero and it shall be considered to have a different reference picture than block E.

– If only one of the three blocks A, B and C has the same reference picture as block E, then the predicted motion vector for block E shall be equal to the motion vector of the A, B, or C block with the same reference picture as block E; otherwise, each component of the predicted motion vector value for block E shall be the median of the corresponding motion vector component values for blocks A, B, and C.

Reference [1] presents further details for special cases when either of the blocks may be outside the picture.

Figure 33 indicates the directional segment predictions for block sizes 16x8 and 8x16. The direction represented by an arrow indicates which neighboring block is used for that sub-block's prediction.

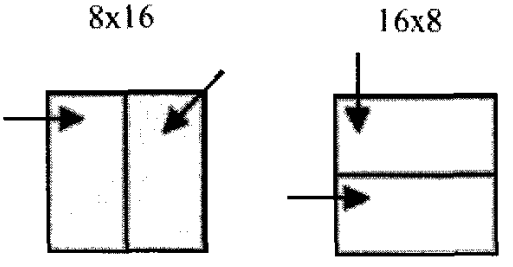


Fig. 33. Directional Segment prediction [21]

If the macroblock where the block to be predicted is coded in 16x8 or 8x16 mode, the prediction is generated as follows (refer to Fig. 32 and the definitions of A, B, C, E above):

a) Motion vector block size 8x16:

1) Left block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used
2) Right block: C is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

b) Motion vector block size 16x8:

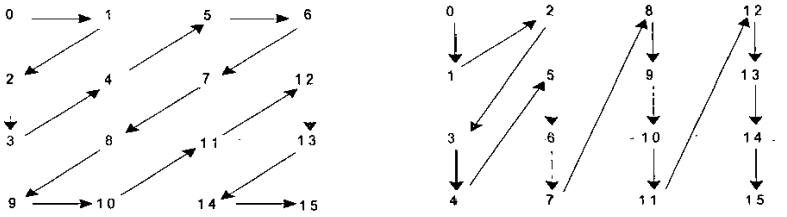
1) Upper block: B is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used
2) Lower block: A is used as prediction if it has the same reference picture as E, otherwise "median prediction" is used

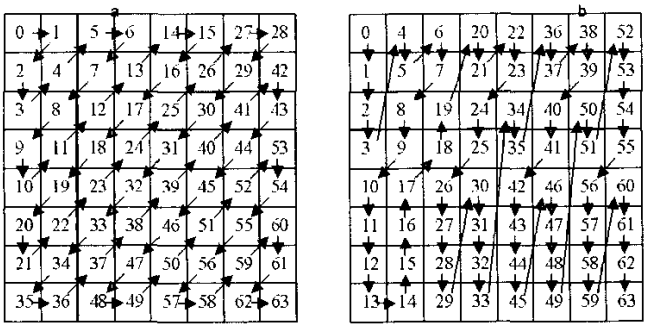
Reference [1] presents further details for special cases when either of the blocks may be outside the picture.

Entropy Coding

Different types of encoding schemes are followed by H.264/MPEG-4 part 10. Universal Variable Length Coding (UVLC) is the default entropy coding. Reference [1] provides a table of code numbers and codewords that can be used. Zig-zag scan is used to read the transform coefficients (Fig. 34 (a)). The table is needed for relation between code number and Level/Run/EOB. The remaining Level/Run combinations are assigned a code number according to the following priority

1. Sign of Level (+-)
2. Run (ascending)
3. Absolute value of Level (ascending)





(a) (b)

Fig. 34. (a) Zig-zag scan [30] **Fig. 34. (b)** Alternate scan [30]

Context-based Adaptive Variable Length Coding (CAVLC) method is used for decoding transform coefficients. The following coding elements are used:

1. If there are non-zero coefficients, it is typically observed that there is a string of coefficients at the highest frequencies that are +1/-1. A common parameter Num-Trail is used that contains the number of coefficients as well as the number of "Trailing 1s". (T1s). For T1s, only the sign has to be decoded.
2. For coefficients other than the T1s, Level information is decoded.
3. Lastly, the Run information is decided. Since the number of coefficients is already known, this limits possible values for Run.

Reference [1] provides three VLC tables for luma and one VLC table for Chroma DC that may be used for Num-Trail coding. It also provides a table to help select the luma tables depending on the number of coefficients in the blocks to the left and above the block under consideration.

Context-based Adaptive Binary Arithmetic Coding (CABAC) provides large bit-rate reduction compared to UVLC-based entropy coding. It involves constructing context models to predict the current symbol under consideration. The non-binary symbols are converted to binary using binary decisions called bins and these bins are then encoded using Binary Arithmetic Coding. A detailed explanation of CABAC is provided in Section 4.2.

Decoding Process

Decoder follows the reverse process of the coder. A macroblock or sub-partition is decoded in the following order.

1. Parsing of syntax elements using VLC/CAVLC or CABAC
2. Inter prediction or Intra prediction
3. Transform coefficient decoding
4. Deblocking filter

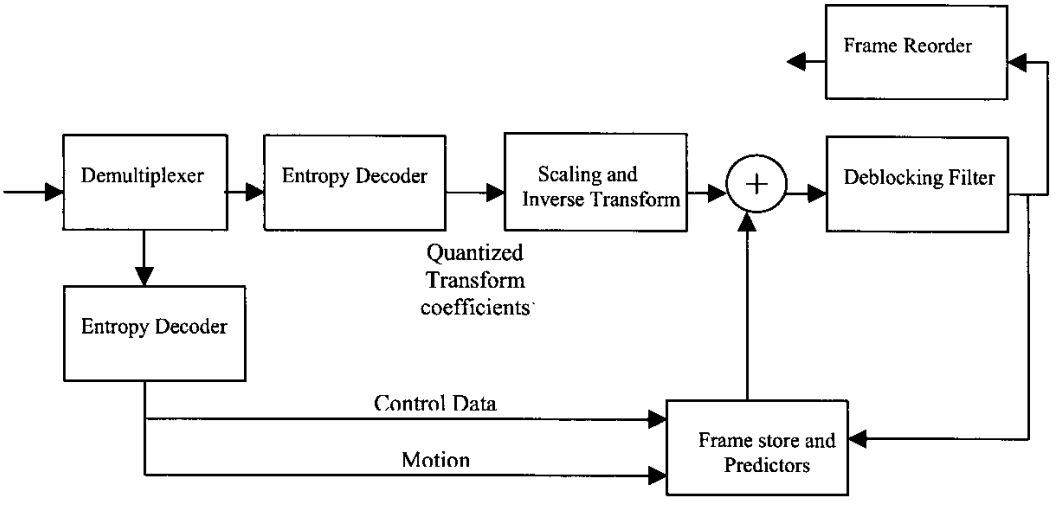


Fig. 35. H.264/MPEG-4 part 10 Decoder

4. SIGNIFICANT FEATURES

4.1 Profiles and Levels

Taking into account that all the users may not require all the features provided by H.264/MPEG-4 part 10, Profiles and Levels have been introduced [21]. It specifies a set of algorithmic features and limits, which shall be supported by all decoders conforming to that profile. The encoders are not required to make use of any particular set of features supported in a profile. For any given profile, Levels generally correspond to processing power and memory capability on a codec. Each level may support a different picture size – QCIF, CIF, ITU-R 601 (SDTV), HDTV, S-HDTV, D-Cinema and data rate varying from a few tens of kilobits per second (kbps) to hundreds of megabits per second (Mbps). Table 6 lists the limits for each level. The non-integer level numbers are referred as "intermediate levels". All levels have the same status, but note that some applications may choose to use only the integer-numbered levels.

4.1.1 Baseline Profile

The profile ID for baseline profile is 66. It supports video conferencing and video telephony applications. The decoders supported by this profile support the following features [21]:

1. I and P slice types
2. De-blocking filter
3. Pictures with field picture flag equal to 0
4. Pictures with alternate scan flag equal to 0
5. Pictures with macroblock adaptive frame field flag equal to 0
6. Zig-zag scan

7. ¼- sample inter prediction
8. Tree-structured motion segmentation down to 4x4 block size
9. VLC-based entropy coding
10. Arbitrary slice order (ASO) – The decoding order of slices within a picture may not follow the constraints that first Macroblock in slice is monotonically increasing within the NAL unit stream for a picture.
11. Flexible macroblock ordering (FMO) (Number of slice groups -1) < 8
    The macroblocks may not necessarily be in the raster scan order. The MAP assigns MBs to a slice group. A maximum of 8 slice groups are allowed.
12. 4:2:0 chroma format
13. Redundant slices – These belong to the redundant coded picture. This picture is not used for decoding unless the primary coded picture is missing or corrupted.

4.1.2 Main Profile

The profile ID for baseline profile is 77. It supports the broadcast video application. The decoders supported by this profile support the following features [21]:

1. B slice type
2. CABAC
3. Adaptive bi-prediction (Weighted prediction)
4. All features included in the baseline except:
   - Arbitrary Slice Order (ASO): In Main profile, the decoding order of slices within a picture follows the constraints that first macroblock in slice is monotonically increasing within the NAL unit stream for a picture.
   - Flexible macroblock ordering (FMO): In Main profile, (Number of slice groups –1) < 0
   - Redundant slices
6. Pictures with field picture flag equal to 1
7. Pictures with macroblock adaptive frame field flag equal to 1
8. Capable of decoding bitstreams confirming to baseline profile if the following additional sequence parameter set constraints are obeyed
   - More_than_one_slice_group_allowed_flag is equal to 0
   - Arbitrary_slice_order_allowed_flag is equal to 0, and
   - Redundant_slices_allowed_flag is equal to 0

4.1.3 Extended profile

The profile ID for baseline profile is 88. The decoders supported by this profile support the following features [21]:

1. B slice type
2. SP and SI slice types
3. Data partitioning slices
4. Adaptive bi-prediction (Weighted prediction)
5. All features included in the Baseline profile
6. Pictures with field picture flag equal to 1
7. Pictures with macroblock adaptive frame field flag equal to 1

All video decoders supporting the Extended profile shall also support the Baseline profile.

Table. 6. Levels and corresponding picture type and frame rate
(SHDTV: Super HDTV, HHR: Horizontal High Resolution, p: progressive, i: interlaced)

Levels	
1.0	QCIF @ 15fps
1.1	QCIF @ 30fps
1.2	CIF @ 15fps
2.0	CIF @ 30fps
2.1	HHR @15 or 30fps
2.2	SDTV @ 15faps
3.0	SDTV: 720x480x20i,720x576x25i 10Mbps(max)
3.1	1280x720x30p, SVGA (800x600) 50+p
3.2	1280x720x60p
4.0	HDTV: 1920x1080x30i, 1280x720x60p, 2kx1kx30p 20Mbps(max)
4.1	HDTV: 1920x1080x30i, 1280x720x60p, 2kx1kx30p 50Mbps(max)
5.0	SHDTV/D-Cinema: 1920x1080x60p, 2.5kx2k..
5.1	SHDTV/D-Cinema: 4kx2k..

Table 7. Parameter set limits for each level
(The non-integer level numbers are referred as "intermediate levels" Entries marked "-" denotes the absence of a corresponding limit Horizontal Motion Vector range does not exceed [-2048,2047.75] luma displacement units) [21]

Level number	Max macroblock processing rate MaxMBPS (MB/s)	Max frame size MaxFS (MBs)	Max decoded picture buffer size MaxDPB (1024 bytes)	Max video bit rate MaxBR (1000 bits/s)	Max CPB size MaxCPB (1000 bits)	Vertical MV component range MaxVmvR (luma frame samples)	Min compression ratio MinCR	Max number of MVs per two consecutive MBs MaxMvsPer2Mb
1	1 485	99	148.5	64	175	[-64,+63.75]	2	-
1.1	3 000	396	337.5	192	500	[-128,+127.75]	2	-
1.2	6 000	396	891.0	384	1 000	[-128,+127.75]	2	-
1.3	11 880	396	891.0	768	2 000	[-128,+127.75]	2	-
2	11 880	396	891.0	2 000	2 000	[-128,+127.75]	2	-
2.1	19 800	792	1 782.0	4 000	4 000	[-256,+255.75]	2	-
2.2	20 250	1 620	3 037.5	4 000	4 000	[-256,+255.75]	2	-
3	40 500	1 620	3 037.5	10 000	10 000	[-256,+255.75]	2	32
3.1	108 000	3 600	6 750.0	14 000	14 000	[-512,+511.75]	4	16
3.2	216 000	5 120	7 680.0	20 000	20 000	[-512,+511.75]	4	16
4	245 760	8 192	12 288.0	20 000	25 000	[-512,+511.75]	4	16
4.1	245 760	8 192	12 288.0	50 000	62 500	[-512,+511.75]	2	16
5	552 960	21 696	40 680.0	135 000	135 000	[-512,+511.75]	2	16
5.1	983 040	36 864	69 120.0	240 000	240 000	[-512,+511.75]	2	16

The following table 8 illustrates the effect of level limits on frame rate for some example picture sizes.

Table 8. Illustration of the effect of level limits on frame rate

Level number:					1	1.1	1.2	1.3	2	2.1	2.2
Max frame size (macroblocks):					99	396	396	396	396	792	1 620
Max macroblocks/second:					1 485	3 000	6 000	11 880	11 880	19 800	20 250
Max picture size (samples):					25 344	101 376	101 376	101 376	101 376	202 752	414 720
Max samples/second:					380 160	768 000	1 536 000	3 041 280	3 041 280	5 068 800	5 184 000
Format	MBs Width	MBs Height	MBs Total	Luma Samples							
SQCIF	128	96	48	12 288	30.9	62.5	125.0	172.0	172.0	172.0	172.0
QCIF	176	144	99	25 344	15.0	30.3	60.6	120.0	120.0	172.0	172.0
QVGA	320	240	300	76 800	-	10.0	20.0	39.6	39.6	66.0	67.5
525 SIF	352	240	330	84 480	-	9.1	18.2	36.0	36.0	60.0	61.4
CIF	352	288	396	101 376	-	7.6	15.2	30.0	30.0	50.0	51.1
525 HHR	352	480	660	168 960	-	-	-	-	-	30.0	30.7
625 HHR	352	576	792	202 752	-	-	-	-	-	25.0	25.6
VGA	640	480	1 200	307 200	-	-	-	-	-	-	16.9
525 4SIF	704	480	1 320	337 920	-	-	-	-	-	-	15.3
525 SD	720	480	1 350	345 600	-	-	-	-	-	-	15.0
4CIF	704	576	1 584	405 504	-	-	-	-	-	-	12.8
625 SD	720	576	1 620	414 720	-	-	-	-	-	-	12.5
SVGA	800	600	1 900	486 400	-	-	-	-	-	-	-
XGA	1024	768	3 072	786 432	-	-	-	-	-	-	-
720p HD	1280	720	3 600	921 600	-	-	-	-	-	-	-
4VGA	1280	960	4 800	1 228 800	-	-	-	-	-	-	-
SXGA	1280	1024	5 120	1 310 720	-	-	-	-	-	-	-
525 16SIF	1408	960	5 280	1 351 680	-	-	-	-	-	-	-
16CIF	1408	1152	6 336	1 622 016	-	-	-	-	-	-	-
4SVGA	1600	1200	7 500	1 920 000	-	-	-	-	-	-	-
1080 HD	1920	1080	8 160	2 088 960	-	-	-	-	-	-	-
2Kx1K	2048	1024	8 192	2 097 152	-	-	-	-	-	-	-
4XGA	2048	1536	12 288	3 145 728	-	-	-	-	-	-	-
16VGA	2560	1920	19 200	4 915 200	-	-	-	-	-	-	-

Level number:					3	3.1	3.2	4	4.1	5	5.1
Max picture size (macroblocks):					1 620	3 600	5 120	8 192	8 192	21 696	36 864
Max macroblocks/second:					40 500	108 000	216 000	245 760	245 760	552 960	983 040
Max picture size (samples):					414 720	921 600	1 310 720	2 097 152	2 097 152	5 554 176	9 437 184
Max samples/second:					10 368 000	27 648 000	55 296 000	62 914 560	62 914 560	141 557 760	251 658 240
Format	MBs Width	MBs Height	MBs Total	Luma Samples							
SQCIF	128	96	48	12 288	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QCIF	176	144	99	25 344	172.0	172.0	172.0	172.0	172.0	172.0	172.0
QVGA	320	240	300	76 800	135.0	172.0	172.0	172.0	172.0	172.0	172.0
525 SIF	352	240	330	84 480	122.7	172.0	172.0	172.0	172.0	172.0	172.0
CIF	352	288	396	101 376	102.3	172.0	172.0	172.0	172.0	172.0	172.0
525 HHR	352	480	660	168 960	61.4	163.6	172.0	172.0	172.0	172.0	172.0
625 HHR	352	576	792	202 752	51.1	136.4	172.0	172.0	172.0	172.0	172.0
VGA	640	480	1 200	307 200	33.8	90.0	172.0	172.0	172.0	172.0	172.0
525 4SIF	704	480	1 320	337 920	30.7	81.8	163.6	172.0	172.0	172.0	172.0
525 SD	720	480	1 350	345 600	30.0	80.0	160.0	172.0	172.0	172.0	172.0
4CIF	704	576	1 584	405 504	25.6	68.2	136.4	155.2	155.2	172.0	172.0
625 SD	720	576	1 620	414 720	25.0	66.7	133.3	151.7	151.7	172.0	172.0
SVGA	800	600	1 900	486 400	-	56.8	113.7	129.3	129.3	172.0	172.0
XGA	1024	768	3 072	786 432	-	35.2	70.3	80.0	80.0	172.0	172.0
720p HD	1280	720	3 600	921 600	-	30.0	60.0	68.3	68.3	153.6	172.0
4VGA	1280	960	4 800	1 228 800	-	-	45.0	51.2	51.2	115.2	172.0
SXGA	1280	1024	5 120	1 310 720	-	-	42.2	48.0	48.0	108.0	172.0
525 16SIF	1408	960	5 280	1 351 680	-	-	-	46.5	46.5	104.7	172.0
16CIF	1408	1152	6 336	1 622 016	-	-	-	38.8	38.8	87.3	155.2
4SVGA	1600	1200	7 500	1 920 000	-	-	-	32.8	32.8	73.7	131.1
1080 HD	1920	1080	8 160	2 088 960	-	-	-	30.1	30.1	67.8	120.5
2Kx1K	2048	1024	8 192	2 097 152	-	-	-	30.0	30.0	67.5	120.0
4XGA	2048	1536	12 288	3 145 728	-	-	-	-	-	45.0	80.0
16VGA	2560	1920	19 200	4 915 200	-	-	-	-	-	28.8	51.2

Please note the following:

- As used in the Table X, "525" refers to typical use for environments using 525 analogue scan lines (of which approximately 480 lines contain the visible picture region), and "625" refers to environments using 625 analogue scan lines (of which approximately 576 lines contain the visible picture region).
- XGA is also known as XVGA, 4SVGA as UXGA, 16XGA as 4Kx3K, CIF as 625 SIF, 625 HHR as 2CIF as half 625 D-1, half 625 ITU-R BT.601, 525 SD as 525 D-1 as 525 ITU-R BT.601, 625 SD as 625 D-1 as 625 ITU-R BT.601.
- Frame rates given are correct for progressive scan modes, and for interlaced if the frame height is divisible by 32.

4.1.3 Adaptive Bi-prediction

If explicit_B_prediction_block_weight_indication is 0, the prediction block of a B-block shall be generated by averaging the pixel values of the prediction blocks. If explicit_B_prediction_indication is 1, the implicit B prediction block weighting is in use. Otherwise, the explicit B prediction block weighting is in use.

Explicit B Prediction Block Weighting

Prediction signal is generated as the following form:

$$P = clip1\left(\frac{P_1 \times FWF + P_2 \times SWF}{2^{LWD}} + D\right)$$

where

P = ABP prediction signal

$$FWF = \begin{cases} -FWFM, & if\ \ FWFS = 0 \\ FWFM, & otherwise \end{cases}$$

$$SWF = \begin{cases} -SWFM, & if\ \ SWFS = 0 \\ SWFM, & otherwise \end{cases}$$

$$D = \begin{cases} -DM, & if\ \ DS = 0 \\ DM, & otherwise \end{cases}$$

P1 = Reference signal corresponding to the reference index ref_idx1
P2 = Reference signal corresponding to the reference index ref_idx2
FWFM = first_weight_factor_magnitude[abp_coeff_idx]
FWFS = first_weight_factor_sign[abp_coeff_idx]
SWFM = second_weight_factor_magnitude[abp_coeff_idx]
SWFS = second_weight_factor_sign[abp_coeff_idx]
DM = constant_factor_magnitude[abp_coeff_idx]
DS = constant_factor_sign[abp_coeff_idx]
LWD = logarithmic_weight_denominator[abp_coeff_idx]

To limit the calculation to 16-bit precision, the following conditions shall be met:

$$-128 \le FWF \le 128$$

$$-128 \le SWF \le 128$$

$$-128 \le FWF + SWF \le 128$$

Implicit B Prediction Block Weighting

Prediction signal generation procedure is specified in the following table. If display order of ref_fwd <= display order of ref_bwd, ABP coefficient (1/2, 1/2, 0) is used. Otherwise, (2, -1, 0) is used. If the reference picture is a long-term picture, its display order is regarded as earlier than all short-term pictures and the picture having the larger default relative index value is regarded as earlier in display order.

ABP coefficient	prediction signal
(1/2, 1/2, 0)	(P1 + P2)/2
(2, -1, 0)	clip1(2*P1 - P2)

(2, -1, 0) coefficients may be used in fading sequences.

4.2 Context-based Adaptive Binary Arithmetic Coding (CABAC)

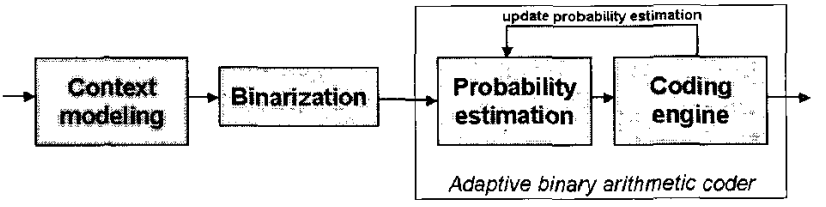


Fig. 36. Schematic Block Diagram for CABAC [13]

A general block diagram for CABAC is shown in Fig. 36 from [13]. The following scheme is used by CABAC [1]:

1. Context models are created based on the neighboring symbols referred to as context modeling. The initialization of context models is explained in the following sub-section.
2. Non-binary symbols are mapped into a sequence of binary decisions called bins.
3. For each bin, a context variable is defined by an equation of prior transmitted symbols. The possible numerical values of a context variable are called contexts and each context has a probability distribution associated with it.
4. The bins are then encoded with Adaptive Binary Arithmetic Coding. After coding of each bin, the probability model is updated using the values of encoded bins.

4.2.1 Initialization of Context models

Initialization involves the probability model associated with each context, the initial counts c_0 and c_1 of the events "0" and "1" respectively. There are three categories of models, based on the initialization:

1. Models with initial counts depending on the quantization parameter (QP)
2. Models with fixed initial counts (independent of the quantization parameter)
3. Models with flat (uniform) initialization

The initial counts after initialization then have to be translated into the representation of the probability models.

The initial counts c_0 and c_1 are given in case of QP –dependent and independent models. QP-dependent models also give the scaling factors r_0 and r_1. These are conditionally rescaled in order to guarantee that the condition $2 <= c_{total} <= 17$ where $c_{total} = c_0 + c_1$. Next, the Most Probable Symbol (MPS) and the *State* corresponding to the underlying probability model is determined by using the rescaled counts of the first step. In the case of models with uniform initialization, the initial values of the counts are given by $c_0 = c_1 = 1$ such that MPS = 0 and *State* = 0.

The standard discusses in detail about the context models and binarization for coding the following:

1. Macroblock modes – the cases of I, P, and B slices, Intra mode
2. Motion information – Motion vector data; reference frame parameters
3. Texture information – Coded Block Pattern (CBP), Intra prediction mode, and transform coefficients.

4.3 Fractional Pel Accuracy

H.264/MPEG-4 part 10 supports one-quarter and one-eighth pel accuracy and the fractional sample accuracy is indicated by a parameter called motion resolution in H.264/MPEG-4 part 10. If motion resolution has value 0, quarter sample resolution with a 6-tap filter is applied to luma samples in the block. If motion resolution value is 1, one-eighth sample interpolation with 8-tap filter is used. The interpolation is equivalent to upsampling of the frame. At the Geneva meeting of JVT, Telenor presented a contribution (COM –16 D.361) [2]. Motion vectors were used with 1/4- and 1/ 8- pel accuracies. Figure 37 shows the interpolation process for the two motion vector accuracies of 1/4 - and 1/8 – pel accuracies [2]. In case of 1/4 - pel Motion Vector (MV) accuracy, the frame has to be upsampled by a factor of 4 where as by a factor of 8 for 1/8- pel MV-accuracy. A combination of filters may be used to upsample by the required factor.

The luma prediction values at half sample positions shall be obtained by applying a 6-tap filter with tap values (1, -5, 20, 20, -5, 1). The luma prediction values at quarter sample positions shall be obtained by averaging samples at integer and half sample positions. Certain mathematical operations involved in this process are explained below:

- >> operator

  The bit-wise operator ">>" is used as "a>>b". a>>b indicates an arithmetic right shift of a two's complement integer representation of *a* by *b* binary digits. This function is defined only for positive values of *b*. Bits shifted into the MSBs (Most Significant Bit) as a result of the right shift shall have a value equal to the MSB of *a* prior to the shift operation.

- Clip1 operator:

$$\text{Clip3}(a, b, c) = \begin{cases} a & ; \quad c < a \\ b & ; \quad c > b \\ c & ; \quad \text{otherwise} \end{cases}$$

$$\text{Clip1}(x) = \text{Clip3}(0, 255, x)$$

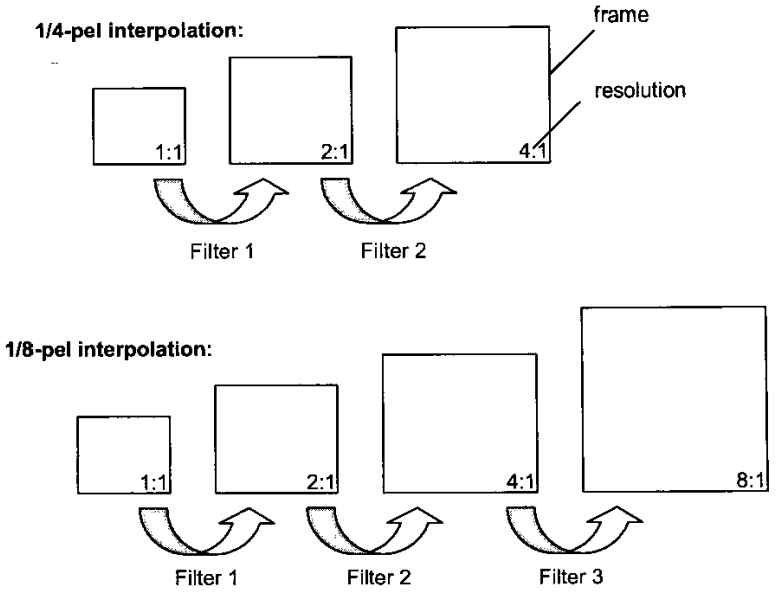


Fig. 37. Illustration of interpolation for fractional pel accuracy [2]

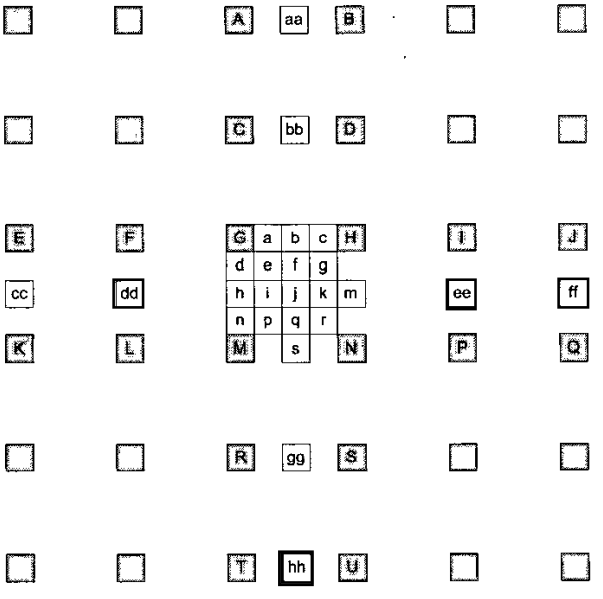


Fig. 38. Integer samples (shaded blocks with upper-case letters) and fractional sample positions (un-shaded blocks with lower-case letters) for quarter luma interpolation [1]

The process for each fractional position is described below.

- The samples at half sample positions labelled 'b' shall be obtained by first calculating intermediate values denoted as '*b*' by applying the 6-tap filter to the nearest integer

position samples in the horizontal direction. The samples at half sample positions labelled 'h' shall be obtained by first calculating intermediate values denoted as '*h*' by applying the 6-tap filter to the nearest integer position samples in the vertical direction:

b = (E-5F+20G+20H-5I+J),
h = (A-5C+20G+20M-5R+T).

The final prediction values shall be calculated using:

b = Clip1((*b*+16)>>5),

h = Clip1((*h*+16)>>5).

- The samples at half sample position labelled as 'j' shall be obtained by first calculating intermediate value denoted as '*j*' by applying the 6-tap filter to the intermediate values of the closest half sample positions in either the horizontal or vertical direction because these yield an equivalent result.

  j = *cc*-5*dd*+20*h*+20*m*-5*ee*+*ff*, or

  j = *aa*-5*bb*+20*b*+20*s*-5*gg*+*hh*,

  where intermediate values denoted as '*aa*', '*bb*', '*gg*', '*s*' and '*hh*' shall be obtained by applying the 6-tap filter horizontally in an equivalent manner to '*b*' and intermediate values denoted as '*cc*', '*dd*', '*ee*', '*m*' and '*ff*' shall be obtained by applying the 6-tap filter vertically in an equivalent manner to '*h*'. The final prediction value shall be calculated using: j = Clip1((*j*+512)>>10).

- The samples at quarter sample positions labelled as 'a', 'c', 'd', 'n', 'f', 'i', 'k' and 'q' shall be obtained by averaging with truncation the two nearest samples at integer and half sample positions using: a = (G+b)>>1, c = (H+b)>>1, d = (G+h)>>1,
  n = (M+h)>>1, f = (b+j)>>1, i = (h+j)>>1, k = (j+m)>>1 and q = (j+s)>>1.

- The samples at quarter sample positions labelled as 'e', 'g', 'p', and 'r' shall be obtained by averaging with truncation the two nearest samples at half sample positions in the diagonal direction using e = (b+h)>>1, g = (b+m)>>1, p = (h+s)>>1, and r = (m+s)>>1.

Chroma sample interpolation

Motion-compensated chroma prediction values at fractional sample positions shall be obtained using the equation below.

$$v = ((s-d^x)(s-d^y)A + d^x(s-d^y)B + (s-d^x)d^yC + d^xd^yD + s^2/2)/s^2$$

where A, B, C and D are the integer position reference picture samples surrounding the fractional sample location; d^x and d^y are the fractional parts of the sample position in units of one eighth samples for quarter sample interpolation or one sixteenth samples for one eighth sample interpolation; and s is 8 for quarter sample interpolation and s is 16 for one eighth sample interpolation. The relationships between the variables in the equation and reference picture positions are illustrated in Fig. 39.

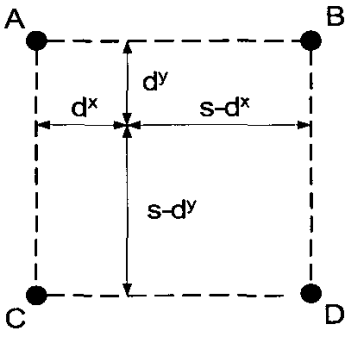


Fig. 39. Fractional sample position dependent variables in chroma interpolation and surrounding integer position samples A, B, C, and D. [1]

Abbreviations are used to specify the filter used. The following rules are used when naming filters. For example, for a 1/8-pel interpolation, a combination of 8-tap 6-tap and 2-tap may be used and is referred to as (862). A study was conducted to analyse the PSNR of the various filter combinations possible for achieving pel accuracy, at different bit rates. Figures 40-47 show the rate-distortion results obtained. Besides the filter combinations, the graphs include

- 1/3- pel with cubic interpolation (TML1)
- AMA- 6 with ½, 1/3, and 1/6 – pel accuracies (TML2)

Rate distortion graphs for ¼ - pel accuracy

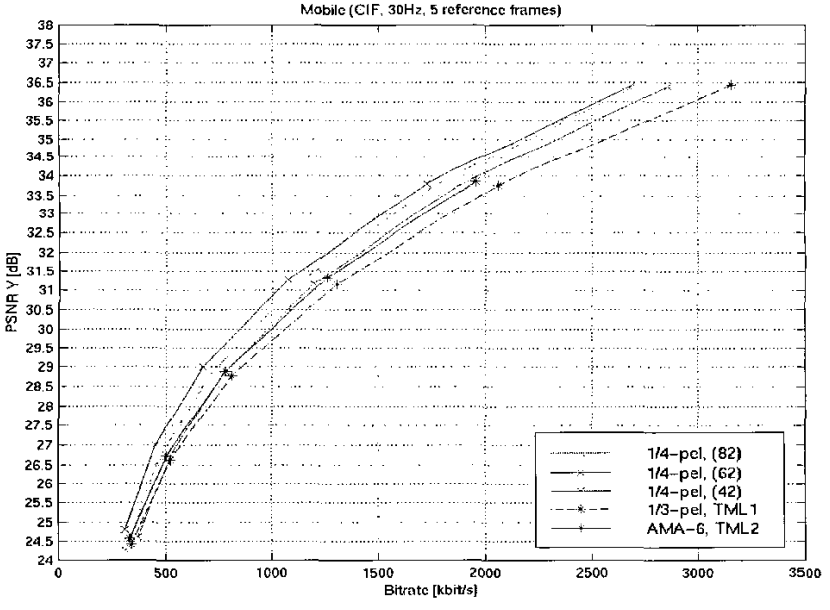


Fig. 40. Rate distortion graph for ¼ - pel accuracy for Mobile CIF sequence [2]

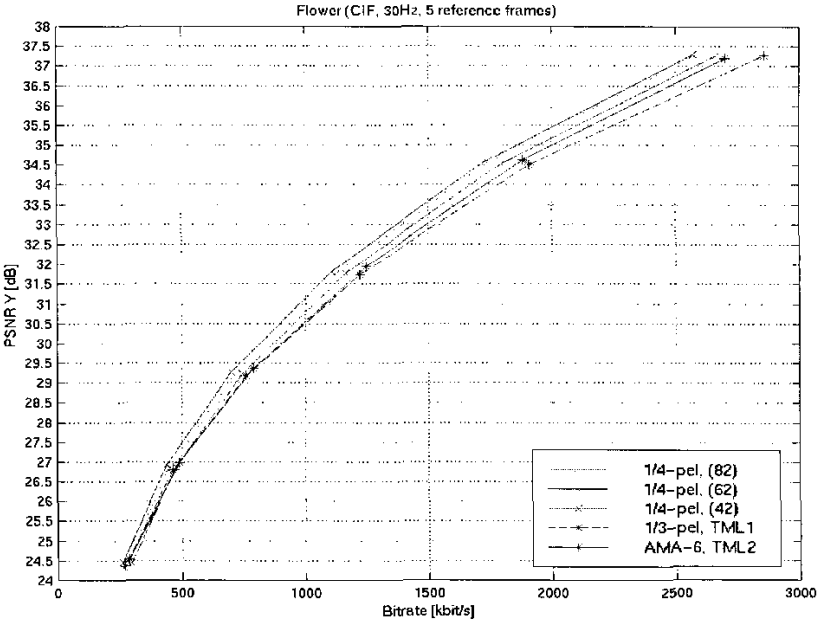


Fig. 41. Rate distortion graph for ¼ - pel accuracy for Flower CIF sequence [2]

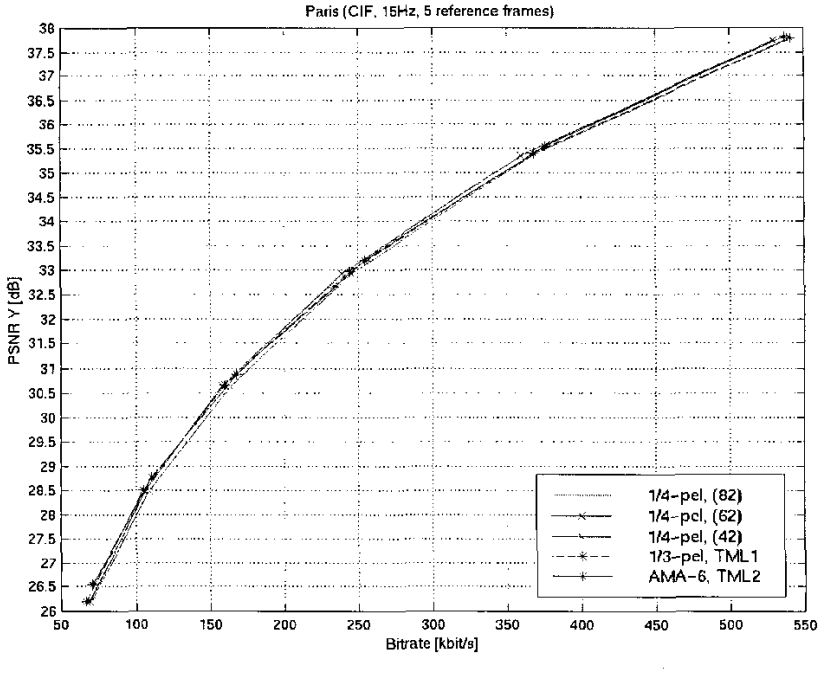


Fig. 42. Rate distortion graph for ¼ - pel accuracy for Paris CIF sequence [2]

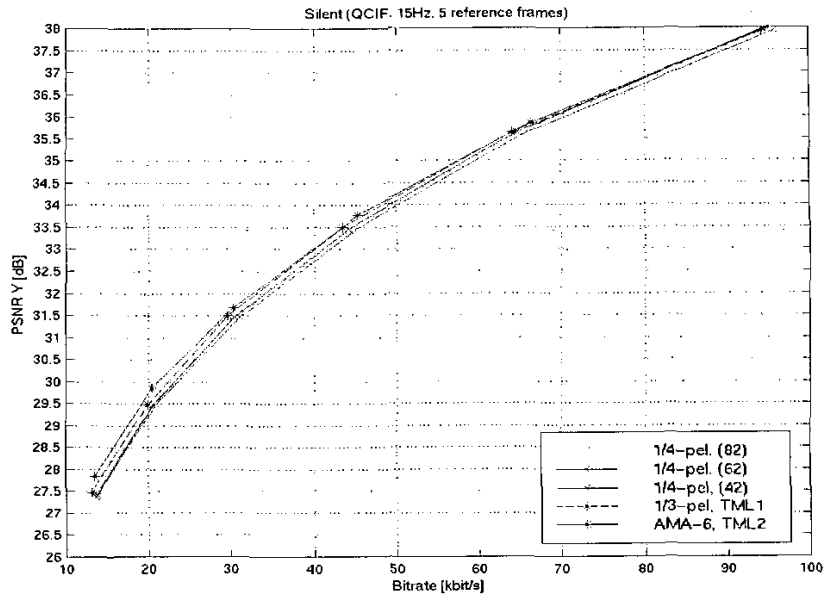


Fig. 43. Rate distortion graph for ¼ - pel accuracy for Silent QCIF sequence [2]

Rate Distortion Graphs for 1/8 Pel Accuracy

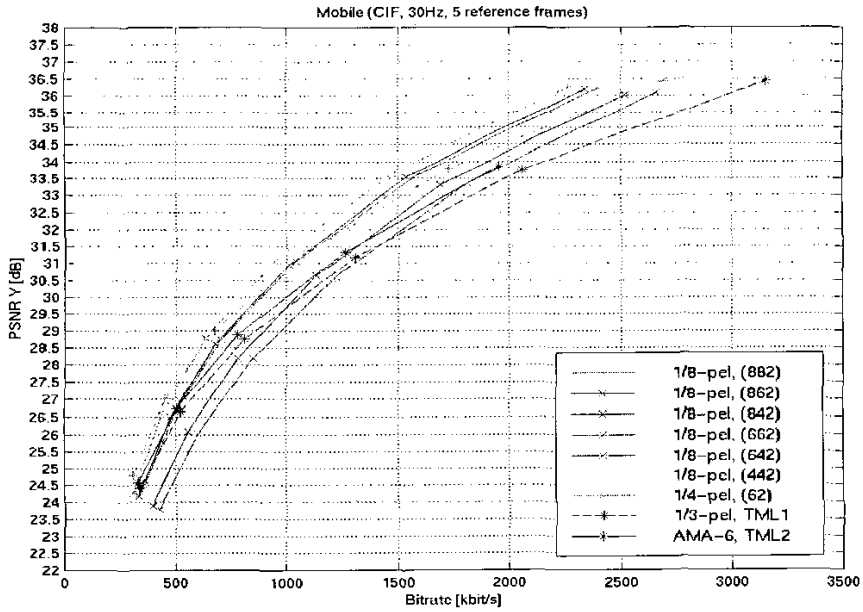


Fig. 44. Rate distortion graph for 1/8 - pel accuracy for Mobile CIF sequence [2]

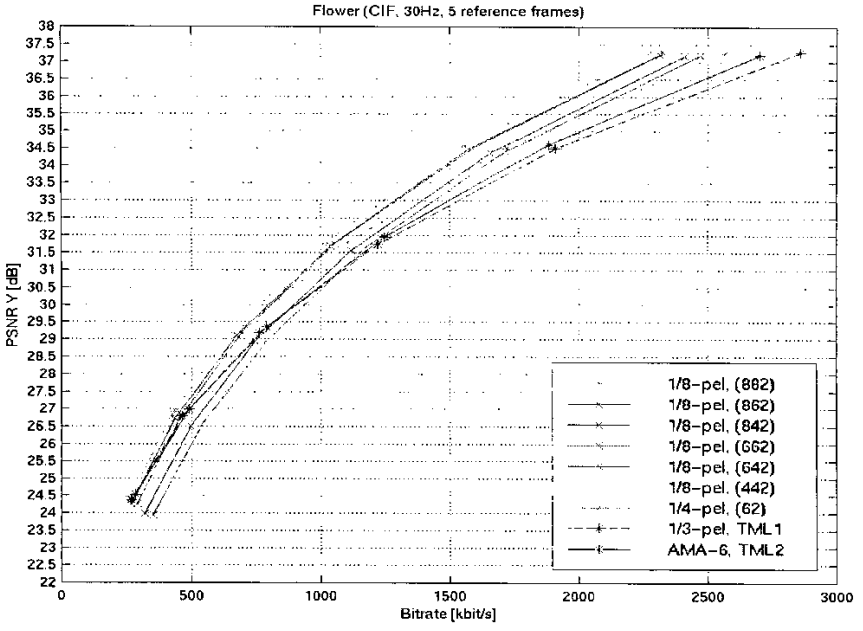


Fig. 45. Rate distortion graph for1/8 - pel accuracy for Flower CIF sequence [2]

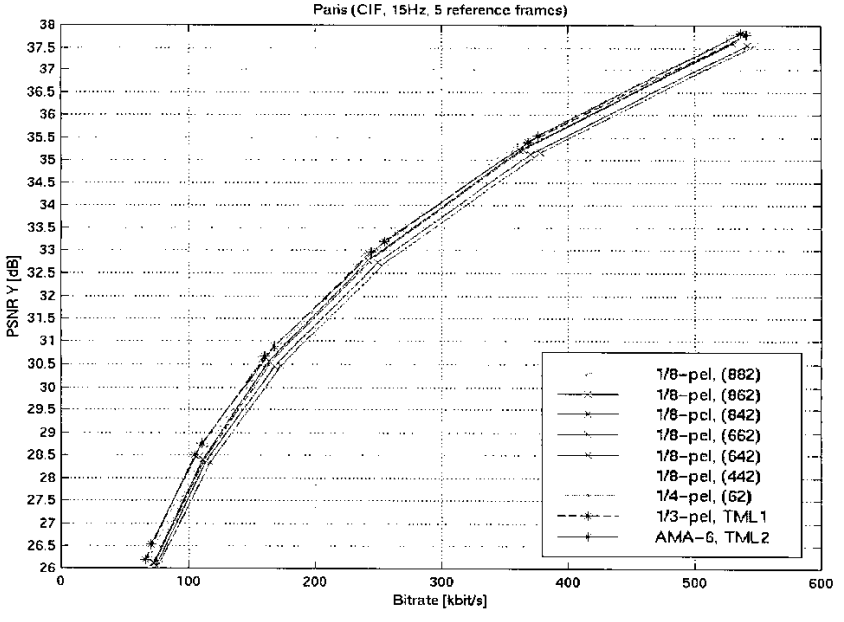


Fig. 46. Rate distortion graph for1/8 - pel accuracy for Paris CIF sequence [2]

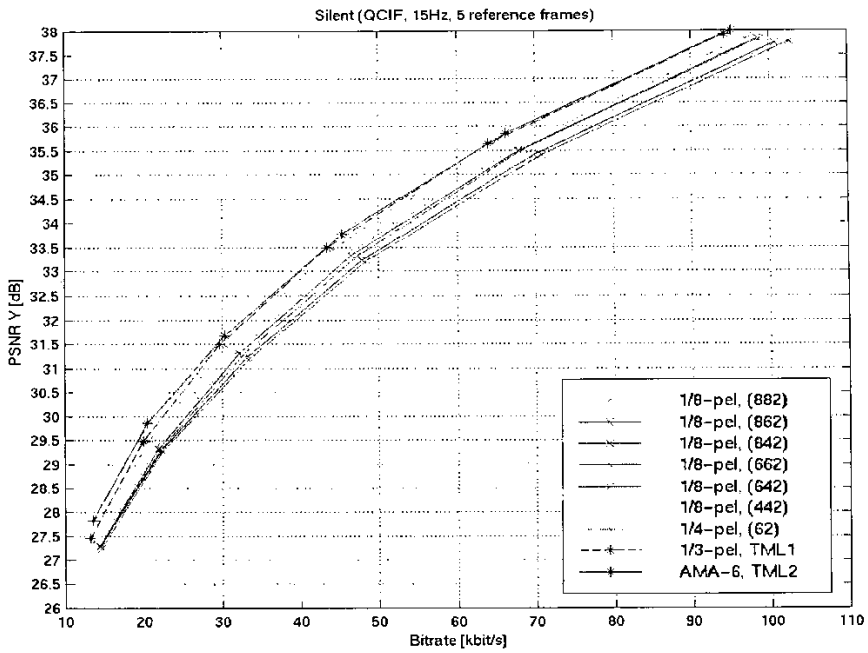


Fig. 47. Rate distortion graph for1/8- pel accuracy for Silent QCIF sequence [2]

Inference from rate distortion graphs

Figures 39-46 show that if only ¼-pel is applied, the combination of 6-tap and bilinear filter (62) seems to be the best choice. If only 1/8-pel is applied, the combination of two 8-tap filters and a bilinear filter (882) seems to be the best choice. For ¼-pel accuracy, 6-tap filter {1,-5,20,20,-5,1} may be used for half-pel values. Quarter pel values are average of integer and half-pels values.

If fixed MV-accuracies are used (no AMA), it is recommended to use the 1/4-pel MV-accuracy with the combination of 6-tap and bilinear filters for the lower data rates and the 1/8-pel MV-accuracy with the combination of two 8-tap and a bilinear filter for the higher data rates instead of fixed 1/3-pel MV accuracy.

4.5. 4x4 Integer Transform

The previous video coding standards relied on Discrete Cosine Transform (DCT) [6] that provided the transformation but produced inverse transform mismatch problems. H.264/MPEG-4 part 10 uses an integer transform with a similar coding gain as a 4x4 DCT. It is multiplier–free, involves additions, shifts in 16-bit arithmetic, thus minimizing computational complexity, especially for low-end processes [7]. The transformation of input pixels $X=\{x_{00} \ldots x_{33}\}$ to output coefficients $Y = \{y_{00} \ldots y_{33}\}$ is defined by: [1]

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}$$

This transform matrix is used in all (except 16x16 Intra DC) the 4x4 block transforms.

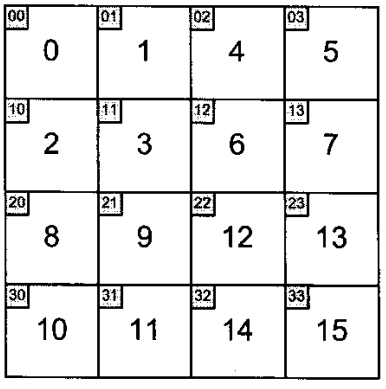


Fig. 48. Assignment of indices for a 4x4 luma block

The 16 luma DC coefficients of 16 (4x4) blocks, are transformed using Walsh Hadamard transform.

$$Y_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_{D00} & x_{D01} & x_{D02} & x_{D03} \\ x_{D10} & x_{D11} & x_{D12} & x_{D13} \\ x_{D20} & x_{D21} & x_{D22} & x_{D23} \\ x_{D30} & x_{D31} & x_{D32} & x_{D33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) // 2$$

Chroma DC coefficients of four 4x4 blocks of each chroma component are transformed using Walsh Hadamard transform.

$$\mathbf{Y} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} DC_{00} & DC_{01} \\ DC_{10} & DC_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

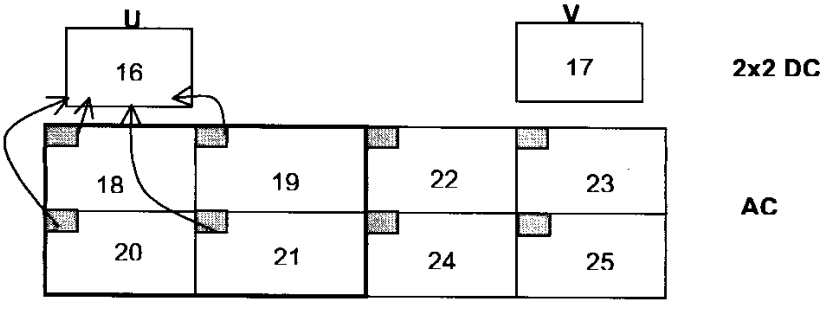


Fig. 49. 8x8 MB – Four 4x4 blocks for each chroma component

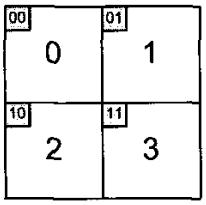


Fig. 50. Assignment of indices for a chroma block corresponding to a 4x4 luma block

Multiplication by two can be performed either through additions or through left shifts, so that no actual multiplication operations are necessary. Thus, the transform is multiplier-free.

For input pixels with 9-bit dynamic range (because they are residuals from 8-bit pixel data), the transform coefficients are guaranteed to fit within 16 bits, even when the second transform for DC coefficients is used. Thus, all transform operations can be computed in 16-bit arithmetic. In fact, the maximum dynamic range of the transform coefficients fills a range of only 15.2 bits; this small headroom can be used to support a variety of different quantization strategies, which are outside the scope of this specification.

The inverse transformation of normalized coefficients $Y'=\{y'_{00} \ldots y'_{33}\}$ to output pixels X' is defined by:

$$X' = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} y'_{00} & y'_{01} & y'_{02} & y'_{03} \\ y'_{10} & y'_{11} & y'_{12} & y'_{13} \\ y'_{20} & y'_{21} & y'_{22} & y'_{23} \\ y'_{30} & y'_{31} & y'_{32} & y'_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

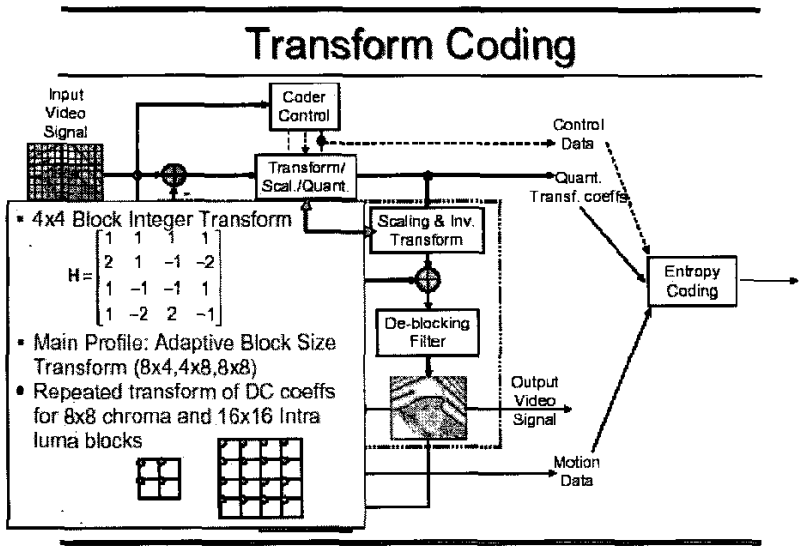


Fig. 51. Block Diagram emphasizing Adaptive Block size Transform [10]

Multiplications by ½ are actually performed via right shifts, so that the inverse transform is also multiplier-free. The small errors introduced by the right shifts are compensated by a larger dynamic range for the data at the input of the inverse transform.

The transform and inverse transform matrices have orthogonal basis functions. Unlike the DCT, though, the basis functions do not have the same norm. Therefore, for the inverse transform to recover the original pixels, appropriate normalization factors must be applied to the transform coefficients before quantization and after de-quantization. Such factors are absorbed by the quantization and de-quantization scaling factors described below. By the above exact definition of the inverse transform, the same operations will be performed on coder and decoder sides. Thus we avoid the usual problem of "inverse transform mismatch".

5. COMPARATIVE STUDIES

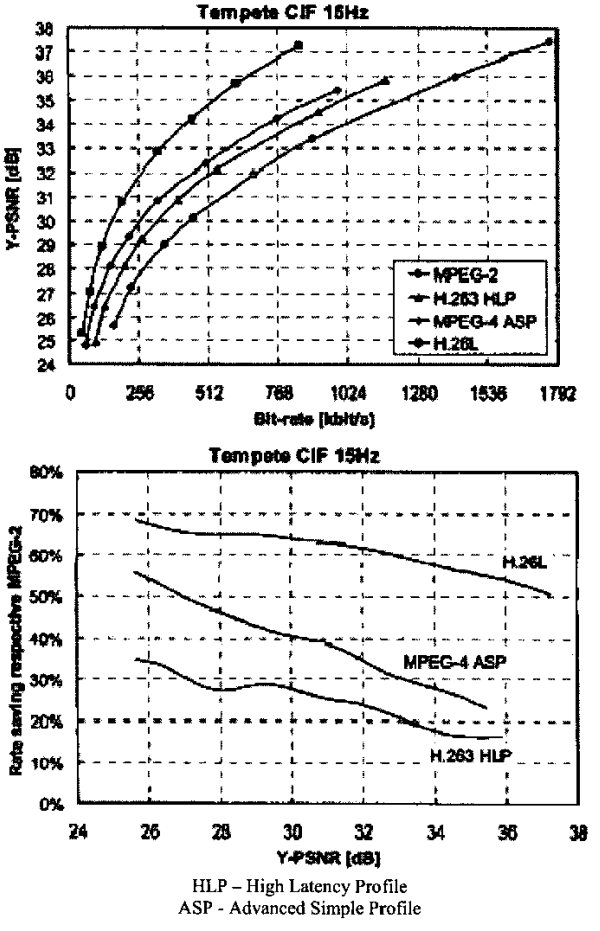
MPEG-2, H.263, MPEG-4 and H.264/MPEG-4 part 10 are based on similar concepts and the main differences involved are the prediction signal, the block sizes used for transform coding and the entropy coding [8]. H.264/MPEG-4 part 10 has some specific additional features, which distinguish it from other standards. H.264/MPEG-4 part 10 uses a more flexible motion compensation model supporting various rectangular partitions in each macroblock. Previous standards allowed only square sized partitions in a macroblock. Multiple reference pictures also help in better prediction, though the complexity is increased. Moreover, quarter-pixel and one-eighth pixel accuracies provide high spatial accuracy. Table 9 provides a comparison of MPEG-1, MPEG-2, MPEG-4 and H.264/MPEG-4 part 10.

Two experiments were performed, first targeting the video streaming and the second targeting the video conferencing. Full search motion estimation with a range of 32 integer pixels was used by all encoders along with the Lagrangian Coder described in [8]. The test sequences consisted of four QCIF sequences coded at 10 Hz and 15 Hz (Foreman, Container, News, Tempete) and four CIF sequences coded at 15 Hz and 30 Hz (Bus, Flower Garden, Mobile and Calendar and Tempete). MPEG-2 with Main Profile and Main Level; H.263 of High-Latency Profile (HLP); MPEG-4 Visual with Advanced Simple Profile (ASP); H.264/MPEG-4 part 10 with ¼ -pel accuracy for QCIF and 1/8 pel accuracy for CIF along with CABAC were used for this experiment. Five reference frames were used for both H.263 and H.264/MPEG-4 part 10 [8].

Table 9. Comparison of standards MPEG-1, MPEG-2, MPEG-4 and H.264/MPEG-4 part 10

Feature/Standard	MPEG-1	MPEG-2	MPEG-4	MPEG-4 part 10/H.264
Macroblock size	16x16	16x16 (frame mode) 16x8 (field mode)	16x16	16x16
Block Size	8x8	8x8	16x16,8x8,16x8 [20]	8x8,8x16,16x8, 16x16,4x8, 8x4,4x4
Transform	DCT	DCT	DCT/Wavelet transform	4x4 integer transform
Transform size	8x8	8x8	8x8	4x4
Quantization step size	Increases with constant increment	Increases with constant increment	Vector quantization used	step sizes that increase at the rate of 12.5%.
Entropy coding	VLC	VLC (different VLC tables for Intra and Inter modes)	VLC	VLC, CAVLC and CABAC
Motion Estimation & Compensation	Yes	Yes	Yes	Yes, More flexible Upto 16 MVs per MB
Pel accuracy	Integer ½-pel	Integer ½-pel	Integer ½-pel ¼-pel	Integer ½-pel ¼ -pel
Profiles	No	5 profiles Several levels within a profile	8 profiles Several levels within a profile	3 profiles Several levels within a profile
Reference frame	Yes One frame	Yes One frame	Yes One frame	Yes Multiple frames (as many as 5 frames allowed)
Picture Types	I, P, B, D	I, P, B	I, P, B	I, P, B, SI, SP
Playback & Random Access	Yes	Yes	Yes	Yes
Error robustness	Synchronization & concealment [19]	Data partitioning, redundancy, FEC for important packet transmission [18]	Synchronization, Data partitioning, Header extension, Reversible VLCs	Deals with packet loss and bit errors in error-prone wireless networks
Transmission rate	Up to 1.5Mbps	2 - 15Mbps	64kbps - ~2Mbps	64kbps - 150Mbps
Encoder complexity	Low	Medium	Medium	High
Compatible with previous standards	Yes	Yes	Yes	No

The PSNR-based results for the video streaming application are as shown in Fig. 52 [8]:



HLP – High Latency Profile
ASP - Advanced Simple Profile

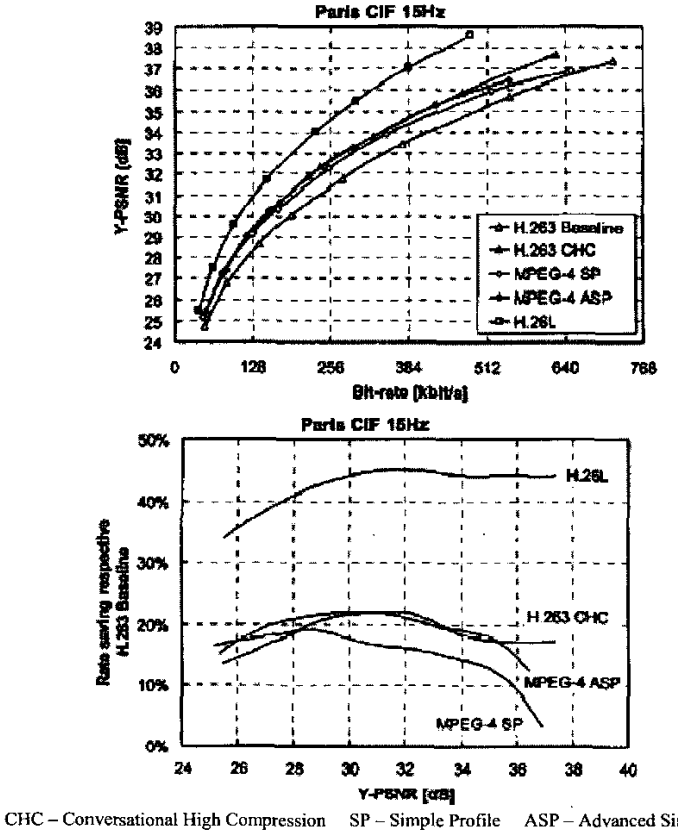
Fig. 52. Rate distortion and bit-rate saving curves for Tempete at 15 Hz in streaming video comparison [8]

Average bit-rate saving for the various standards show that H.26L/MPEG-4 part 10 provided more than 35% bit-rate saving relative to MPEG-4 ASP and H.263.

Table 10. Average bit rate savings for video streaming [8]

	Average bit-rate savings relative to		
Coder	MPEG-4 ASP	H.263	MPEG-2
H.264/MPEG-4 part 10	39%	49%	64%
MPEG-4 ASP	-	17%	43%
H.263 HLP	-	-	31%

The PSNR-based results for the second experiment with video conferencing application were are shown in Fig. 53 [8]. The test sequences consisted of four QCIF sequences encoded at 10 Hz and 15 Hz (Akiyo, Foreman, Mother and Daughter, Silent Voice) and four CIF sequences encoded at 15 Hz and 30 Hz (Carphone, Foreman, Paris, Sean). H.263 Baseline, Conversational High Compression (CHC); MPEG-4 Simple Profile (SP), ASP and JVT/H.26L were used for this experiment.



CHC – Conversational High Compression SP – Simple Profile ASP – Advanced Simple Profile

Fig. 53. Rate-distortion and bit-rate saving curves for Paris at 15 Hz in conversational video comparison [8]

Table 11 provides average bit-rate savings of standards MPEG-4 ASP, H.263 CHC, MPEG-4 SP, H.263 Baseline relative to each other. It shows that JVT/H.26L provided more than 40% bit-rate saving relative to H.263 Baseline and 25% relative to MPEG-4 ASP and H.263 CHC.

Table 11. Average bit-rate savings for conversational use [8]

	Average bit-rate savings relative to			
Coder	MPEG-4 ASP	H.263 CHC	MPEG-4 SP	H.263 Baseline
JVT	28%	32%	34%	45%
MPEG-4 ASP	-	7%	10%	24%
H.263 CHC	-	-	2%	18%
MPEG-4 SP	-	-	-	16%

5. IMPLEMENTATIONS

Some companies have demonstrated real-time preliminary prototypes [10]. Please note that these are incomplete implementations of non-final draft designs.

- UB Video (JVT-C148) demonstrated CIF Resolution operation [22]
- Videolocus (JVT-D023) demonstrated SDTV resolution [23]
- Sand Video [33] demonstrated its high definition H.264 video decoder at the 2003 Consumer Electronics Shows in Las Vegas
- Envivio [32] and Nagravision [35] demonstrated their Digital Rights Management (DRM) and Conditional Access solution based on Internet Streaming Media Alliance at the Broadband Plus Show in December 2002
- Philips [37] launched products like Nexperia pnx1500, Platform4 Media Adapter and Platform4 Player Micro Edition which are compatible with H.264/MPEG-4 part 10
- Polycom [38] announced an implementation of H.264/MPEG-4 part 10 on a video system and a multipoint control unit for video conferencing and collaboration
- PixtelTools Corporation Inc. announced the release of software encoder based on H.264 [43]

Other companies progressing towards implementation [10] are:

- HHI (HEINRICH-HERTZ-INSTITUT) [24]
- Deutsche Telekom [25]
- Broadcom [26]
- Nokia [27]
- Motorola [28]
- Harmonic Inc [36]
- Microsoft [31] and Intel [39] announced in Jan. 2003 to work together to develop a portable media player (PMP) hardware reference design for the new Microsoft Media2Go software platform, including high-performance video software for H.264/MPEG 4 Part 10 video codec.
- VCON [41]
- KMV Technologies [44]

6. FUTURE WORK

Reference [10] lists the following future work

- Standard Systems and file format support specifications
- IPR licensing
- Industry interpretative testing of implementations
- Verification testing by standard organizations to measure capabilities of the new standard (final tests by July 2003)
- Standardizing reference software implementation (end of 2003 - early 2004)
- Standardizing conformance bitstreams and specifications (end of 2003 – early 2004)
- Standardizing example encoding description (end of 2003 – early 2004)
- Consideration of potential extensions (more than 8 bits per sample, 4:4:4 sampling format etc.)

7. CONCLUSIONS

H.264/MPEG-4 part 10 has very good features like multiple reference frames, CABAC, different profiles and is suitable for applications like video streaming and video conferencing. Error resilience is achieved by using parameter sets, which can be either transmitted in-band or out-of band. It is more flexible compared to the previous standards and this enables improved coding efficiency. However, it should be noted that this is at the expense of added complexity to the coder/decoder. Also, it is not backward compatible to the previous standards. The level of complexity of the decoder is reduced by designing it specifically for a profile and a level. H.264/MPEG-4 part 10 provides three profiles and levels within them. In all, H.264/MPEG-4 part 10 seems to have a combination for good video storage and real-time video applications. UB Video has been developing a complete video processing solution that is based on H.264/MPEG-4 part 10 and is optimized for the Texas Instruments TMS320C64x digital media platform family of DSPs called UBLive–26L-C64 [11]. It represents a software/hardware combination solution to most high-performance video applications. The bandwidth is reduced by as much as 50% compared to H.263 and this indicates an important factor for the standard's quick acceptance. Various other companies are developing software/hardware products based on this emerging standard.

ACKNOWLEDGEMENTS

The authors acknowledge with thanks the power point slides and standard documentation provided by Dr. Gary Sullivan, JVT Rapporteur | Chair, ITU-T Rapporteur of Advanced Video Coding. The authors also thank Ms. Hemamalini Narayanan, University of Texas at Arlington for taking out time for discussions and providing guidance. The authors also extend their gratitude towards Dr. Yasser Syed and CableLabs for sharing the latest developments at the JVT meetings.

APPENDIX A

Appendix A.1 (References) provides a list of IEEE papers, conference documents and books used for writing this overview. Further work is actively being done in this area. The Appendix A.2 (Additional Information) provides details about topics not covered in detail in this paper. Please note that the "Additional Information" list is not comprehensive.

A.1 REFERENCES

[1] Joint Video Team (JVT), "Editor's Proposed Modifications to Joint Committee Draft (CD) of Joint Video Specification" (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 4th JVT meeting, Klagenfurt, Austria, 22-26 July, 2002.

[2] T. Wedi, "Interpolation filters for motion compensated prediction with 1/4 and 1/8-pel accuracy", ITU-T Q.15/SG16, doc. Q15-J-14, Osaka, Japan, May 2000.

[3] T. Stockhammer, M. Hannuksela and S. Wenger, "H.26L/JVT Coding Network Abstraction Layer and IP-based Transport", Special session "The Emerging

JVT/H.26L Video coding standard", ICIP 2002, Rochester, New York, vol. 2, pp. 485-488, Sept. 2002.

[4] J. Ribas-Corbera, P. A. Chou and S. Regunathan, "A Flexible Decoder Buffer Model for JVT Video Coding", Special session "The Emerging JVT/H.26L Video coding standard", ICIP 2002, Rochester, New York, vol. 2, pp. 493-496, Sept. 2002.

[5] http://www.itu.int/ITU-T/news/jvtpro.html

[6] K. R. Rao and J. J. Hwang, "Techniques and Standards for Image, Video and Audio Coding", Upper Saddle River, NJ: Prentice Hall, 1996.

[7] H. Malvar et al, "Low – Complexity Transform and Quantization with 16-bit Arithmetic for H.26L", Special session "The Emerging -JVT/H.26L Video coding standard", ICIP 2002, Rochester, New York, vol. 2, pp. 489-492, Sept. 2002.

[8] A. Joch et al, "Performance Comparison of Video Coding Standards using Lagrangian Coder Control", Special session "The Emerging JVT/H.26L Video coding standard", ICIP 2002, Rochester, New York, vol. 2, pp. 501-504, Sept. 2002.

[9] D. Tian et al, "Coding of Faded Scene Transitions", Special session "The Emerging JVT/H.26L Video coding standard", ICIP 2002, Rochester, New York, vol. 2, pp. 505-508, Sept. 2002.

[10] G. J. Sullivan, Plenary Speaker, "Advances in Video compression and emerging JVT/H.26L/AVC standard", ICIP 2002, Rochester, New York, Sept. 2002.

[11] "Emerging H.26L Standard: Overview and TMS320C64x Digital Media Platform Implementation", White Paper, UB Video Inc. [21]

[12] http://standards.pictel.com/ftp/video-site/0201_Gen/JVT-B067.doc

[13] http://www.stanford.edu/class/ee398b/handouts/09-VideoCodingStandards.pdf

[14] B. Girod and M. Flierl, "Multi-frame Motion Compensated Video Compression for the Digital Set-top Box", Invited paper, ICIP 2002, Rochester, New York, vol. 2, pp. 1-4, Sept. 2002.

[15] Joint Video Team, "JVT IPR Status Report, Joint Video team of ISO/IEC MPEG and ITU-T VCEG (ISO/IEF JTC1/SC29/WG11 and ITU-T SG16 Q.6)", 2nd meeting at Geneva, Jan. 29-Feb. 1, 2002.

[16] Joint Video Team, Joint Model Number 1, Revision 1(JM-1r1), JVT – A003r1 document, Pattaya, Thailand, Dec. 2001.

[17] T. Wiegand, "Editor's Proposed modifications to Joint Committee Draft (CD) of Joint Video Specification (ITU Rec. H.264 | ISO/IEC 14496-10 AVC) relative to JVT-D015d5", JVT-D017 draft 0, Klagenfurt, Austria, 22-26 July, 2002.

[18] http://www.eurasip.org/phd_abstracts/frossard-pascal.htm

[19] I. E. G. Richardson, "Video Codec Design", Baffins Lane, Chichester, West Sussex, PO019, IUD, England, Wiley, pp. 64-75, 2002.

[20] http://article.gmane.org/gmane.comp.video.uci.devel/20

[21] Joint Video Team (JVT), "Study of Final Committee Draft of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Draft 2", Filename: JVT-F100d2.doc, 6th Meeting: Awaji, Island, JP, 5-13 December, 2002.

[22] UBVideo website http://www.ubvideo.com/mainmenu.html

[23] VideoLocus website http://www.videolocus.com

[24] HHI website http://www.hhi.de/

[25] Deutsche Telekom website http://www.telekom.de//dtag/home/portal/0,14925,E,00.html

[26] Broadcom website http://www.broadcom.com

[27] Nokia website http://nokia.com

[28] Motorola website http://www.motorola.com
[29] JVT ftp website ftp://ftp.imtc-files.org/jvt-experts
[30] Joint Video Team, "New Interlace Coding Tools", Joint Video team of ISO/IEC MPEG and ITU-T VCEG (ISO/IEF JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-B068, 2nd meeting at Geneva, Jan. 29-Feb. 1, 2002.
[31] Microsoft website: http://microsoft.com/
[32] Envivio website: http://www.envivio.com/
[33] Joint Video Team, "Working Draft Number 2, Revision 2" JVT-B118r2, 2nd meeting at Geneva, Jan. 29-Feb. 1, 2002.
[34] Sand Video website: http://www.sandvideo.com
[35] Nagravision website: www.nagravision.com
[36] Harmonic Inc. website http://www.harmonicinc.com
[37] Philips website: http://www.philips.com
[38] Polycom website: http://www.polycom.com
[39] Intel website: http://www.intel.com
[40] Ajay Luthra, "MPEG-4 AVC/ ITU-T H.264 An Overview", Motorola, Feb. 28, 2002
[41] VCON website: http://www.vcon.com
[42] Vanguard Software Solution Inc. website: http://www.vsofts.com
[43] PixtelTools Corporation Inc. website: http://www.pixteltools.com
[44] KMV Technologies http://www.kmvtechnologies.com

A.2 ADDITIONAL INFORMATION

[1] A. Bovik, (ED), " Handbook of image and video processing," Orlando, FL: Academic Press, 2000.
[2] J. Miano, " Compressed image file formats: JPEG, PNG, GIF, XBM, BMP" Boston, MA: Addison Wesley, 2000.
[3] G. Stockman and L.G. Shapiro, " Computer vision," Upper Saddle River, NJ: Prentice Hall, 2001.
[4] Y. Wang, Y-Quin and J. Ostermann, " Video processing and communications" Upper Saddle River, NJ: Prentice Hall, 2001.
[5] M.T. Sun and A.R. Reibman, " Compressed video over networks," New York, NY: Marcel Dekker, 2000.
[6] Y.Q. Shi and H. Sun, " Image and video compression for multimedia engineering: Fundamentals, algorithms and standards," Boca Raton, FL: CRC Press, 2000.
[7] W.K Pratt, " Digital image processing: PIKS inside" New York, NY: Wiley, 2001.
[8] J. Chen, U.V. Koc and K.J.R. Liu, " Design of digital video coding systems" New York, NY: Marcel Dekker, 2001.
[9] L. Guan, S.Y. Kung and J. Larsen, " Multimedia image and video processing," Boca Raton, FL: CRC Press, 2001.
[10] B. Furht, " Handbook of internet and multimedia systems and applications," Boca Raton, FL: CRC Press, 1999.
[11] J.C. Russ, " The image processing handbook," Boca Raton, FL: CRC Press, 2002.
[12] Special issue on transform coding, IEEE SP Magazine, vol. 18, Sept. 2001.
[13] S.W. Perry, H.S. Wong and L. Guan, " Adaptive image processing," Boca Raton, FL: CRC Press, 2002.

[14] K. R. Rao, Z. S. Bojkovic and D. A. Milovanovic, " Multimedia communication systems," Upper Saddle River, NJ: Prentice-Hall, 2002.
[15] P. Yip and K.R. Rao, " The fast transform and data compression handbook," Boca Raton, FL: CRC Press, 2001.
[16] I.E.G. Richardson, " Video codec design," New York, NY: Wiley, 2002.
[17] B.S. Manjunath, P. Salembier and T. Sikora, " Introduction to MPEG-7," New York, NY: Wiley, 2002.
[18] K. Konstantinides et al, " Design of an MPEG-2 codec," IEEE SP Magazine, vol.19, pp.32-41, July 2002.
[19] Y. Wang, J. Ostermann and Y.Q. Zhang, " Video processing and communications," Upper Saddle River, NJ: Prentice-Hall, 2002.
[20] D. Hankerson et al, " Information theory and data compression," II Edition, Boca Raton, FL: CRC Press, 2003.
[21] D. Hankerson and G.A. Harris, " Transform methods and image compression: An introduction to JPEG and wavelet transform techniques using Octave and Matlab," Linux Journal, pp. 18-24, Jan.1999.
[22] ftp://ftp.imtc-files.org All documents related to JVT (H.264 & MPEG-4 Part 10)
[23] G. Held and T.R. Marshall, "Data compression: techniques and applications, Hardware and software considerations, " New York, NY: Wiley, 1994.
[24] D. Solomon, " Data compression: the complete reference," II Edition, Verlag Heidelberg, Germany: Springer, 2000.
[25] K. Sayood, "Introduction to Data Compression" San Francisco, CA: Morgan Kaufmann, II Edition, 2000.
[26] B. Furht, S. W. Smoliar and H. J. Zhang, "Video and Image Processing in Multimedia Systems" Norwell, MA: Kluwer, 1995
[27] D. Marpe et al, "Context-based adaptive binary arithmetic coding in JVT/H.26L", ICIP 2002, Rochester, New York, vol. 2, pp. 513-516, Sept. 2002.
[28] T. Stockhammer, T. Wiegand and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks", ICIP 2002, Rochester, New York, vol. 2, pp. 173-176, Sept. 2002.
[29] R. Kureeren and M. Karzcewicz, "Synchronization-predictive coding for video compression: the SP frames design for JVT/H.26L", ICIP 2002, Rochester, New York, vol. 2, pp. 497-500, Sept. 2002.
[30] G. Sullivan, JVT Chair, garysull@microsoft.com
[31] A. Luthra, JVT Co-chair, aluthra@motorola.com
[32] T. Wiegand, JVT Co-chair, wiegand@hhi.de
[33] Joint Video Team reflector subscription http://mail.imtc.org/cgi-bin/lyris.pl?enter=jvt-experts
[34] Joint Video Team reflector, jvt-experts@mail.imtc.org

APPENDIX B

This provides a list of http and ftp sites that provide video databases, products, vendors etc.

[1] ftp://ftp.imtc-files.org/jvt-experts/
[2] http://standards.pictel.com/ftp/video-site/
[3] www.cablelabs.com

[4] ftp://standard.pictel.com/video-site/h26L/
[5] ftp://standard.pictel.com/ftp/video-site/0201_Gen/
[6] http://ise.stanford.edu/video.html
[7] www.netvideo.com
[8] http://sipi.usc.edu/services/database/Database.html
[9] http://www.mpeg.org
[10] http://www.signalogic.com/mp3.htm
[11] http://mambo.ucsc.edu/psl/olivetti.html
[12] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[13] http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html
[14] www.powerweb.de/mpeg/mpegfaq
[15] www.image.cityu.edu.hk/mspid/index.html
[16] www.image.cityu.edu.hk/imagedb/
[17] ftp://ipl.rpi.edu/pub/image/still/usc/gray/
[18] www.xray.hmc.psu.edu/dicom
[19] http://jpg0.terra.vein.hu/testimages

LIST OF ACRONYMS

Abbreviations	Expansion
3G	Third Generation
ABT	Adaptive Block size Transform
AC	Alternating Current
AMA	Adaptive Motion vector Accuracy
ASO	Arbitrary Slice Order
ASP	Advanced Simple Profile
AVC	Advanced Video Coding
BMP	Bit Map Picture
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CBP	Coded Block Pattern
CHC	Conversational High Compression
CIF	Common Intermediate Format
CPB	Coded Picture Buffer
DC	Direct Current
DCT	Discrete Cosine Transform
DPB	Decoded Picture Buffer
DSL	Digital Subscriber Line
EOB	End Of Block
FMO	Flexible Macroblock Ordering
Fps	Frames per second
GOP	Group Of Pictures

HLP	High Latency Profile
HRD	Hypothetical Reference Decoder
ID	Identity
IEC	International Electrotechnical Commission
IP	Internet Protocol
IPR	Intellectual Property Rights
ISO	International Organization for Standards
ITU-T	International Telecommunications Union - Telecommunication Standardization Sector
JVT	Joint Video Team
MAP	Macroblock Allocation Map
MB	Macroblock
MPEG	Moving Picture Experts Group
MPS	Most Probable Symbol
MSE	Mean Square Error
MV	Motion Vector
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
RTP	Real-time Transport Protocol
SDTV	Standard Definition Television
SP	Simple Profile
T1	Trailing 1s
TML	Test Model Long term
UDP	User Datagram Protocol
UVLC	Universal Variable Length Coding
VBR	Variable Bit Rate
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
WM	Weighting Matrix