Feature

Video coding with H.264/AVC: Tools, Performance, and Complexity

Jörn Ostermann, Jan Bormans, Peter List,
Detlev Marpe, Matthias Narroschke,
Fernando Pereira, Thomas Stockhammer, and Thomas Wedi

Abstract

H.264/AVC, the result of the collaboration between the ISO/IEC Moving Picture Experts Group and the ITU-T Video Coding Experts Group, is the latest standard for video coding. The goals of this standardization effort were enhanced compression efficiency, network friendly video representation for interactive (video telephony) and non-interactive applications (broadcast, streaming, storage, video on demand). H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards. Compared to previous standards, the decoder complexity is about four times that of MPEG-2 and two times that of MPEG-4 Visual Simple Profile. This paper provides an overview of the new tools, features and complexity of H.264/AVC.

Index Terms—H.263, H.264, JVT, MPEG-1, MPEG-2, MPEG-4, standards, video coding, motion compensation, transform coding, streaming

1. Introduction

The new video coding standard Recommendation H.264 of ITU-T also known as International Standard 14496-10 or MPEG-4 part 10 Advanced Video Coding (AVC) of ISO/IEC [1] is the latest standard in a sequence of the video coding standards H.261 (1990) [2], MPEG-1 Video (1993) [3], MPEG-2 Video (1994) [4], H.263 (1995, 1997) [5], MPEG-4 Visual or part 2 (1998) [6]. These previous standards reflect the technological progress in video compression and the adaptation of video coding to different applications and networks. Applications range from video telephony (H.261) to consumer video on CD (MPEG-1) and broadcast of standard definition or high definition TV (MPEG-2). Networks used for video communications include switched networks such as PSTN (H.263, MPEG-4) or ISDN (H.261) and packet networks like ATM (MPEG-2, MPEG-4), the Internet (H.263, MPEG-4) or mobile networks (H.263, MPEG-4). The importance of new network access technologies like cable modem, xDSL, and UMTS created demand for the new video coding standard H.264/AVC, providing enhanced video compression performance in view of interactive applications like video telephony requiring a low latency system and non-interactive applications like storage, broadcast, and streaming of standard definition TV where the focus is on high coding efficiency. Special consideration had to be given to the performance when using error prone networks like mobile channels (bit errors) for UMTS and GSM or the Internet (packet loss) over cable modems, or xDSL. Comparing the H.264/AVC video coding tools like multiple reference frames, $^1/_4$ pel motion compensation, deblocking filter or integer transform to the tools of previous video coding standards, H.264/AVC brought in the most algorithmic discontinuities in the evolution of standardized video coding. At the same time, H.264/AVC achieved a leap in coding performance that was not foreseen just five years ago. This progress was made possible by the video experts in ITU-T and MPEG who established the Joint Video Team (JVT) in December 2001 to develop this H.264/AVC video coding standard. H.264/AVC was finalized in March 2003 and approved by the ITU-T in May 2003. The corresponding standardization documents are downloadable from ftp://ftp.imtc-files.org/jvt-experts and the reference software is available at http://bs.hhi.de/~suehring/tml/download.

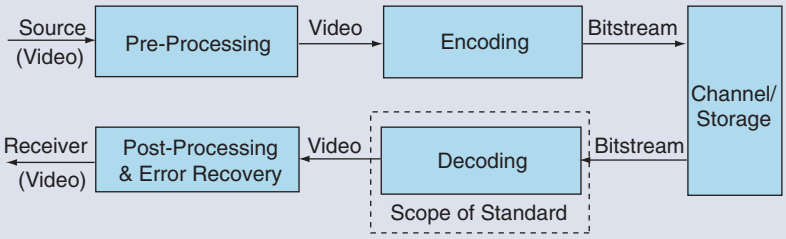


Figure 1. Scope of video coding standardization: Only the syntax and semantics of the bitstream and its decoding are defined.

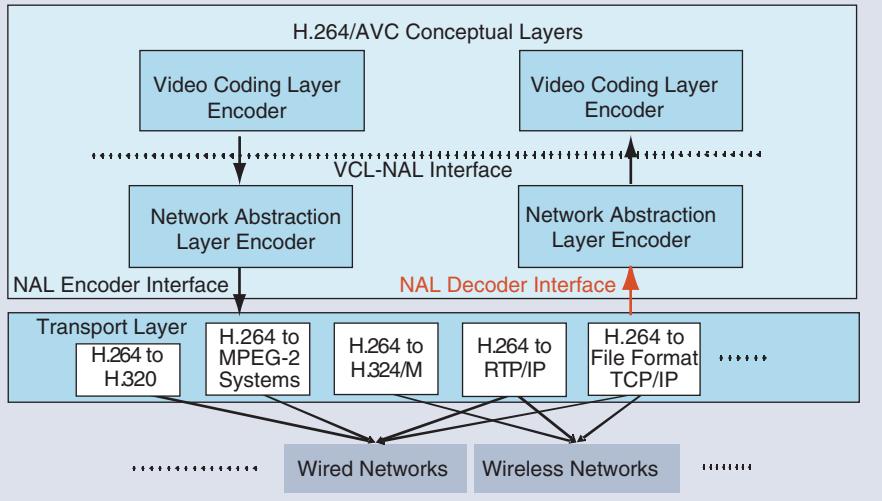


Figure 2. H.264/AVC in a transport environment: The network abstraction layer interface enables a seamless integration with stream and packet-oriented transport layers (from [7]) .

Modern video communication uses digital video that is captured from a camera or synthesized using appropriate tools like animation software. In an optional pre-processing

Jörn Ostermann is with the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, Hannover, Germany. Jan Bormans is with IMEC, Leuven, Belgium. Peter List is with Deutsche Telecom, T-Systems, Darmstadt, Germany. Detlev Marpe is with the Fraunhofer-Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany. Matthias Narroschke is with the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, Appelstr. 9a, 30167 Hannover, Germany, narrosch@tnt.uni-hannover.de. Fernando Peirera is with Instituto Superior Técnico - Instituto de Telecomunicações, Lisboa, Portugal. Thomas Stockhammer is with the Institute for Communications Engineering, Munich University of Technology, Germany. Thomas Wedi is with the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, Hannover, Germany.

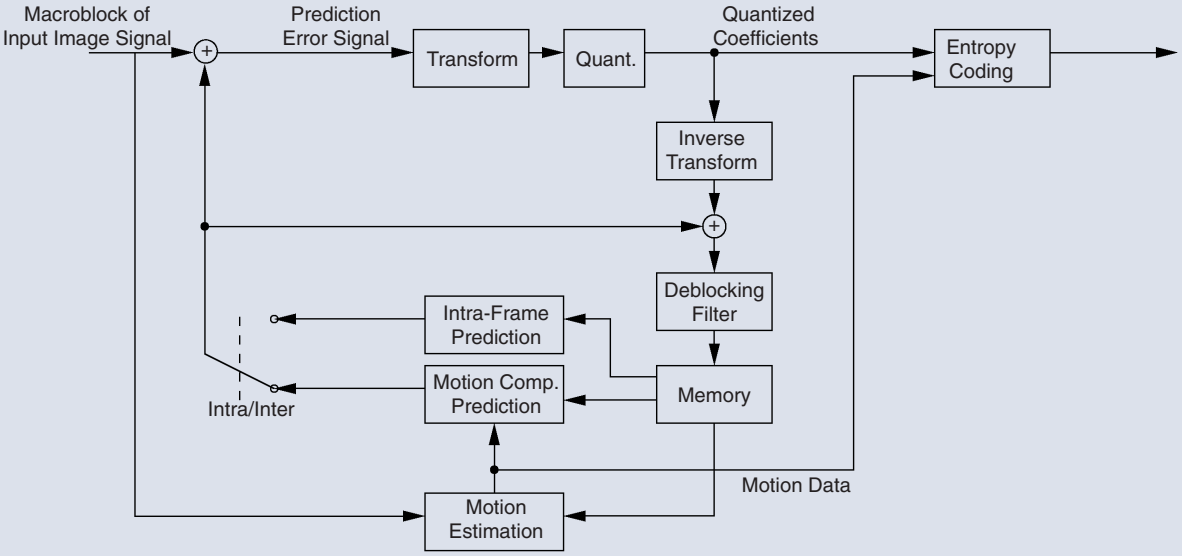


Figure 3. Generalized block diagram of a hybrid video encoder with motion compensation: The adaptive deblocking filter and intra-frame prediction are two new tools of H.264.

step (Figure 1), the sender might choose to preprocess the video using format conversion or enhancement techniques. Then the encoder encodes the video and represents the video as a bit stream. After transmission of the bit stream over a communications network, the decoder decodes the video which gets displayed after an optional post-processing step which might include format conversion, filtering to suppress coding artifacts, error concealment, or video enhancement.

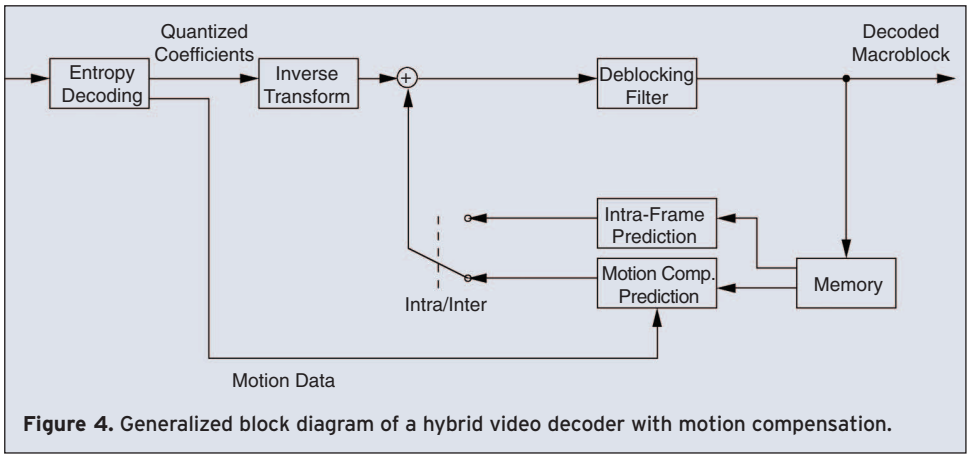


Figure 4. Generalized block diagram of a hybrid video decoder with motion compensation.

The standard defines the syntax and semantics of the bit stream as well as the processing that the decoder needs to perform when decoding the bit stream into video. Therefore, manufactures of video decoders can only compete in areas like cost and hardware requirements. Optional post-processing of the decoded video is another area where different manufactures will provide competing tools to create a decoded video stream optimized for the targeted application. The standard does not define how encoding or other video pre-processing is performed thus enabling manufactures to compete with their encoders in areas like cost, coding efficiency, error resilience and error recovery, or hardware requirements. At the same time, the standardization of the bit stream and the decoder preserves the fundamental requirement for any communications standard—interoperability.

For efficient transmission in different environments not only coding efficiency is relevant, but also the seamless and easy integration of the coded video into all current and future protocol and network architectures. This includes the public Internet with best effort delivery, as well as wireless networks expected to be a major application for the new video coding standard. The adaptation of the coded video representation or bitstream to different transport networks was typically defined in the systems specification in previous MPEG standards or separate standards like H.320 or H.324. However, only the close integration of network adaptation and video coding can bring the best possible performance of a video communication system. Therefore H.264/AVC consists of two conceptual layers (Figure 2). The video coding layer (VCL) defines the efficient representation of the video, and the network adaptation layer (NAL) converts the VCL repre-

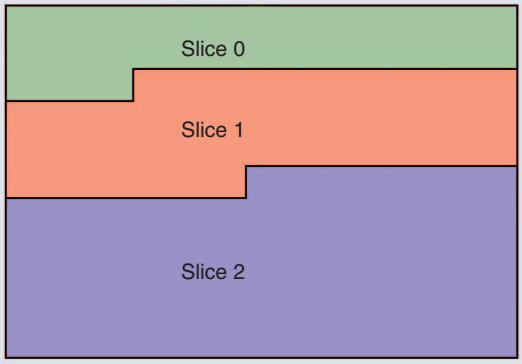


Figure 5. Partitioning of an image into several slices.

sentation into a format suitable for specific transport layers or storage media. For circuit-switched transport like H.320, H.324M or MPEG-2, the NAL delivers the coded video as an ordered stream of bytes containing start codes such that these transport layers and the decoder can robustly and simply identify the structure of the bit stream. For packet switched networks like RTP/IP or TCP/IP, the NAL delivers the coded video in packets without these start codes.

This paper gives an overview of the working, performance and hardware requirements of H.264/AVC. In Section 2, the concept of standardized video coding schemes is introduced. In Section 3, we describe the major tools of H.264/AVC that achieve this progress in video coding performance. Video coder optimization is not part of the standard. However, the successful use of the encoder requires knowledge on encoder control that is presented in Section 4. H.264/AVC may be used for different applications with very different constraints like computational resources, error resilience and video resolution. Section 5 describes the profiles and levels of H.264/AVC that allow for the adaptation of the decoder complexity to different applications. In Section 6, we give comparisons between H.264/AVC and previous video coding standards in terms of coding efficiency as well as hardware complexity. H.264/AVC uses many international patents, and Section 7 paraphrases the current licensing model for the commercial use of H.264/AVC.

2. Concept of Standardized Video Coding Schemes

Standardized video coding techniques like H.263, H.264/AVC, MPEG-1, 2, 4 are based on hybrid video coding. Figure 3 shows the generalized block diagram of such a hybrid video encoder.

The input image is divided into macroblocks. Each macroblock consists of the three components Y, Cr and Cb. Y is the luminance component which represents the brightness information. Cr and Cb represent the color information. Due to the fact that the human eye system is less sensitive to the chrominance than to the luminance the chrominance signals are both subsampled by a factor of 2 in horizontal and vertical direction. Therefore, a macroblock consists of one block of 16 by 16 picture elements for the luminance component and of two blocks of 8 by 8 picture elements for the color components.

These macroblocks are coded in Intra or Inter mode. In Inter mode, a macroblock is predicted using motion compensation. For motion compensated prediction a displacement vector is estimated and transmitted for each block (motion data) that refers to the corresponding position of its image signal in an already transmitted reference image stored in memory. In Intra mode, former standards set the prediction signal to zero such that the image can be coded without reference to previously sent information. This is important to provide for error resilience and for entry points into the bit streams enabling random access. The prediction error, which is the difference between the original and the predicted block, is transformed, quantized and entropy coded. In order to reconstruct the same image on the decoder side, the quantized coefficients are inverse transformed and added to the prediction signal. The result is the reconstructed macroblock that is also available at the decoder side. This macroblock is stored in a memory. Macroblocks are typically stored in raster scan order.

With respect to this simple block diagram (Figure 3), H.264/AVC introduces the following changes:

1. In order to reduce the block-artifacts an adaptive deblocking filter is used in the prediction loop. The deblocked macroblock is stored in the memory and can be used to predict future macroblocks.
2. Whereas the memory contains one video frame in previous standards, H.264/AVC allows storing multiple video frames in the memory.
3. In H.264/AVC a prediction scheme is used also in Intra mode that uses the image signal of already transmitted macroblocks of the same image in order to predict the block to code.
4. The Discrete Cosine Transform (DCT) used in former standards is replaced by an integer transform.

Figure 4 shows the generalized block diagram of the corresponding decoder. The entropy decoder decodes the quantized coefficients and the motion data, which is used for the motion compensated prediction. As in the encoder, a prediction signal is obtained by intra-frame or motion compensated prediction, which is added to the inverse transformed coefficients. After deblocking filtering, the macroblock is completely decoded and stored in the memory for further predictions.

In H.264/AVC, the macroblocks are processed in so called slices whereas a slice is usually a group of macroblocks processed in raster scan order (see Figure 5). In special cases, which will be discussed in Section 3.6, the

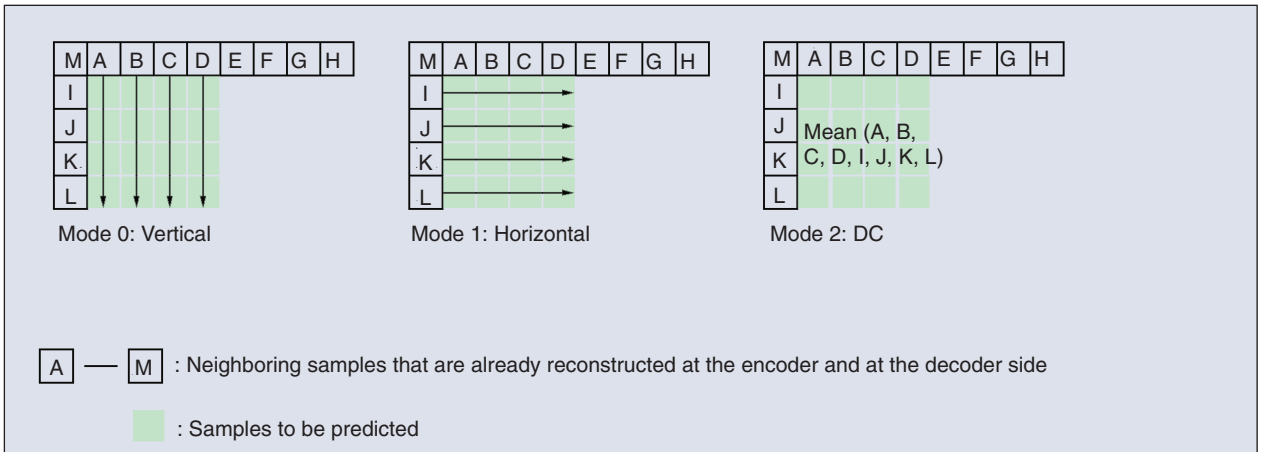


Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

processing can differ from the raster scan order. Five different slice-types are supported which are I-, P-, B-, SI,- and SP-slices. In an I-slice, all macroblocks are encoded in Intra mode. In a P-slice, all macroblocks are predicted using a motion compensated prediction with one reference frame and in a B-slice with two reference frames. SI- and SP-slices are specific slices that are used for an efficient switching between two different bitstreams. They are both discussed in Section 3.6.

For the coding of interlaced video, H.264/AVC supports two different coding modes. The first one is called *frame mode*. In the frame mode, the two fields of one frame are coded together as if they were one single progressive frame. The second mode is called *field mode*. In this mode, the two fields of a frame are encoded separately. These two different coding modes can be selected for each image or even for each macroblock. If they are selected for each image, the coding is referred to *as picture adaptive field/frame coding* (P-AFF). Whereas MPEG-2 allows for selecting the frame/field coding on a macroblock level H.264 allow for selecting this mode on a vertical macroblock pair level. This coding is referred to *as macroblock-adaptive frame/field coding* (MB-AFF). The choice of the frame mode is efficient for regions that are not moving. In non-moving regions there are strong statistical dependencies between adjacent lines even though these lines belong to different fields. These dependencies can be exploited in the frame mode. In the case of moving regions the statistical dependencies between adjacent lines are much smaller. It is more efficient to apply the field mode and code the two fields separately.

3. The H.264/AVC Coding Scheme

In this Section, we describe the tools that make H.264 such a successful video coding scheme. We discuss Intra coding, motion compensated prediction, transform coding, entropy coding, the adaptive deblocking filter as well as error robustness and network friendliness.

3.1 *Intra Prediction*

Intra prediction means that the samples of a macroblock are predicted by using only information of already transmitted macroblocks of the same image. In H.264/AVC, two different types of intra prediction are possible for the prediction of the luminance component Y.

The first type is called INTRA_4×4 and the second one INTRA_16×16. Using the INTRA_4×4 type, the macroblock, which is of the size 16 by 16 picture elements (16×16), is divided into sixteen 4×4 subblocks and a prediction for each 4×4 subblock of the luminance signal is applied individually. For the prediction purpose, nine different prediction modes are supported. One mode is DC-prediction mode, whereas all samples of the current 4×4 subblock are predicted by the mean of all samples neighboring to the left and to the top of the current block and which have been already reconstructed at the encoder and at the decoder side (see Figure 6, Mode 2). In addition to DC-prediction mode, eight prediction modes each for a specific prediction direction are supported. All possible directions are shown in Figure 7. Mode 0 (vertical prediction) and Mode 1 (horizontal prediction) are shown explicitly in Figure 6. For example, if the vertical prediction mode is applied all samples below sample A (see Figure 6) are predicted by sample A, all samples below sample B are predicted by sample B and so on.

Using the type INTRA_16×16, only one prediction mode is applied for the whole macroblock. Four different prediction modes are supported for the type INTRA_16×16: Vertical prediction, horizontal prediction, DC-prediction and plane-prediction. Hereby plane-prediction uses a linear function between the neighboring samples to the left and to the top in order to predict the current samples. This mode works very well in areas of a gently changing luminance. The mode of operation of these modes is the same as the one of the 4×4 prediction modes. The only difference is that they are applied for the whole macroblock instead of for a 4×4 subblock. The effi-

ciency of these modes is high if the signal is very smooth within the macroblock.

The intra prediction for the chrominance signals Cb and Cr of a macroblock is similar to the INTRA_16×16 type for the luminance signal because the chrominance signals are very smooth in most cases. It is performed always on 8×8 blocks using vertical prediction, horizontal prediction, DC-prediction or plane-prediction. All intra prediction modes are explained in detail in [1].

3.2 *Motion Compensated Prediction*

In case of motion compensated prediction macroblocks are predicted from the image signal of already transmitted reference images. For this purpose, each macroblock can be divided into smaller partitions. Partitions with luminance block sizes of 16×16, 16×8, 8×16, and 8×8 samples are supported. In case of an 8×8 sub-macroblock in a P-slice, one additional syntax element specifies if the corresponding 8×8 sub-macroblock is further divided into partitions with block sizes of 8×4, 4×8 or 4×4 [8]. The partitions of a macroblock and a sub-macroblock are shown in Figure 8.

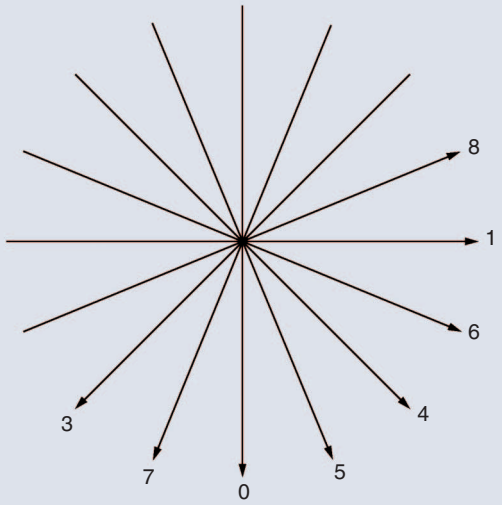


Figure 7. Possible prediction directions for INTRA_4×4 mode.

In former standards as MPEG-4 or H.263, only blocks of the size 16×16 and 8×8 are supported. A displacement vector is estimated and transmitted for each block, refers to the corresponding position of its image signal in an already transmitted reference image. In former MPEG standards this reference image is the most recent preceding image. In H.264/AVC it is possible to refer to several preceding images. For this purpose, an additional picture reference parameter has to be transmitted together with the motion vector. This technique is denoted as motion-compensated prediction with multiple reference frames [9]. Figure 9 illustrates the concept that is also extended to B-slices.

The accuracy of displacement vectors is a quarter of a picture element (quarter-pel or $^1/_4$-pel). Such displacement vectors with fractional-pel resolution may refer to positions in the reference image, which are spatially located between the sampled positions of its image signal. In order to estimate and compensate fractional-pel displacements, the image signal of the reference image has to be generated on sub-pel positions by interpolation. In H.264/AVC the luminance signal at half-pel positions is generated by applying a one-dimensional 6-tap FIR filter, which was designed to reduce aliasing components that deteriorate the interpolation and the motion compensated prediction [8]. By averaging the luminance signal at integer- and half-pel positions the image signal at quarter-pel positions is generated. The chrominance signal at all fractional-pel positions is obtained by averaging.

In comparison to prior video-coding standards, the classical concept of B-pictures is extended to a generalized B-slice concept in H.264/AVC. In the classical concept, B-pictures are pictures that are encoded using both past and future pictures as references. The prediction is obtained by a linear combination of forward and backward prediction signals. In former standards, this linear combination is just an averaging of the two prediction signals whereas H.264/AVC allows *arbitrary weights*. In this generalized concept, the linear combination of prediction signals is also made regardless of the temporal direction.

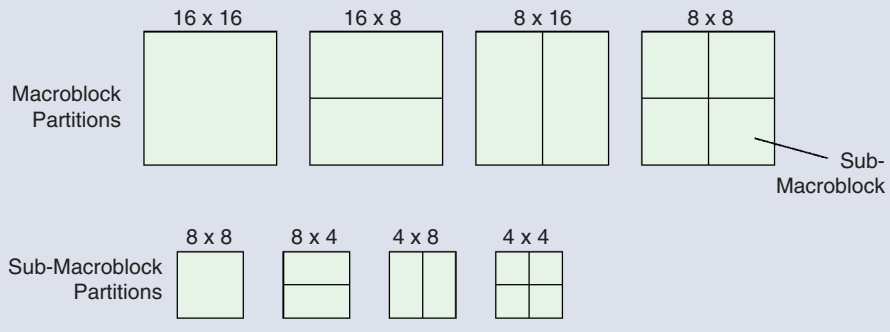


Figure 8. Partitioning of a macroblock and a sub-macroblock for motion compensated prediction.

For example, a linear combination of two forward-prediction signals may be used (see Figure 9). Furthermore, using H.264/AVC it is possible to use images containing B-slices as reference images for further predictions which was not possible in any former standard. Details on this generalized B-slice concept, which is also known as multi-hypothesis motion-compensated prediction can be found in [10], [11], [12].

3.3 *Transform Coding*

Similar to former standards transform coding is applied in order to code the prediction error signal. The task of the transform is to reduce the spatial redundancy of the prediction error signal. For the purpose of transform coding, all former standards such as MPEG-1 and MPEG-2 applied a two dimensional Discrete Cosine Transform (DCT) [13] of the size 8×8. Instead of the DCT, different integer transforms are applied in H.264/ AVC. The size of these transforms is mainly 4×4, in special cases 2×2. This smaller block size of 4×4 instead of 8×8 enables the encoder to better adapt the prediction error coding to the boundaries of moving objects, to match the transform block size with the smallest block size of the motion compensation, and to generally better adapt the transform to the local prediction error signal.

Three different types of transforms are used. The first type is applied to all samples of all prediction error blocks of the luminance component Y and also for all blocks of both chrominance components Cb and Cr regardless of whether motion compensated prediction or intra prediction was used. The size of this transform is 4×4. Its transform matrix H_1 is shown in Figure 10.

If the macroblock is predicted using the type INTRA_16×16, the second transform, a Hadamard transform with matrix H_2 (see Figure 10), is applied in addition to the first one. It transforms all 16 DC coefficients of the already transformed blocks of the luminance signal. The size of this transform is also 4×4.

The third transform is also a Hadamard transform but of size 2×2. It is used for the transform of the 4 DC coefficients of each chrominance component. Its matrix H_3 is shown in Figure 10.

The transmission order of all coefficients is shown in Figure 11. If the macroblock is predicted using the intra prediction type INTRA_16×16 the block with the label "−1" is transmitted first. This block contains the DC coefficients of all blocks of the luminance component. Afterwards all blocks labeled "0"–"25" are transmitted whereas blocks "0"–"15" comprise all AC coefficients of the blocks of the luminance component. Finally, blocks "16" and "17" comprise the DC coefficients and blocks "18"–"25" the AC coefficients of the chrominance components.

Compared to a DCT, all applied integer transforms have only integer numbers ranging from −2 to 2 in the transform matrix (see Figure 10). This allows computing the transform and the inverse transform in 16-bit arithmetic using only low complex shift, add, and subtract operations. In the case of a Hadamard transform, only add and subtract operations are necessary. Furthermore, due to the exclusive use of integer operations mismatches of the inverse transform are completely avoided which was not the case in former standards and caused problems.

All coefficients are quantized by a scalar quantizer. The quantization step size is chosen by a so called quantization parameter QP which supports 52 different quantization parameters. The step size doubles with each increment of 6 of QP. An increment of QP by 1 results in an increase of the required data rate of approximately 12.5%. The transform is explained in detail in [15].

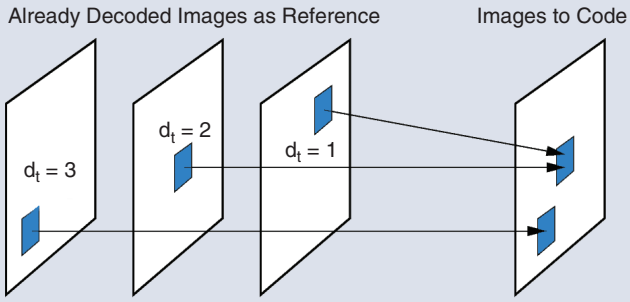


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 10. Matrices H_1, H_2 and H_3 of the three different transforms applied in H.264/AVC.

3.4 *Entropy Coding Schemes*

H.264/AVC specifies two alternative methods of entropy coding: a low-complexity technique based on the usage of context-adaptively switched sets of variable length codes, so-called CAVLC, and the computationally more demanding algorithm of context-based adaptive binary arithmetic coding (CABAC). Both methods represent major improvements in terms of coding efficiency com-

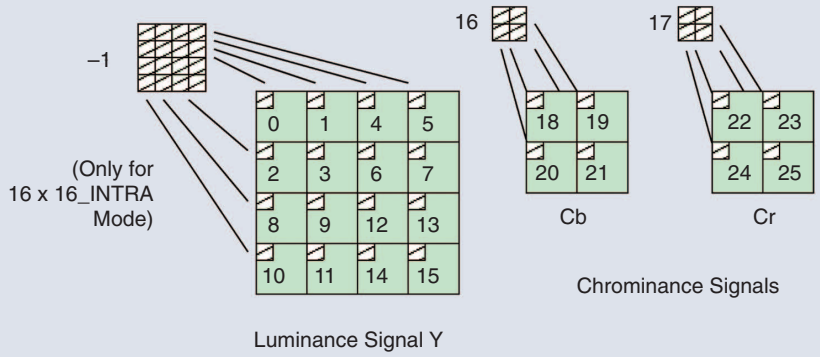


Figure 11. Transmission order of all coefficients of a macroblock [14].

pared to the techniques of statistical coding traditionally used in prior video coding standards. In those earlier methods, specifically tailored but fixed variable length codes (VLCs) were used for each syntax element or sets of syntax elements whose representative probability distributions were assumed to be closely matching. In any case, it was implicitly assumed that the underlying statistics are stationary, which however in practice is seldom the case. Especially residual data in a motion-compensated predictive coder shows a highly non-stationary statistical behavior, depending on the video content, the coding conditions and the accuracy of the prediction model. By incorporating context modeling in their entropy coding framework, both methods of H.264/AVC offer a high degree of adaptation to the underlying source, even though at a different complexity-compression trade-off.

CAVLC is the baseline entropy coding method of H.264/AVC. Its basic coding tool consists of a single VLC of structured Exp-Golomb codes, which by means of individually customized mappings is applied to all syntax elements except those related to quantized transform coefficients. For the latter, a more sophisticated coding scheme is applied. As shown in the example of Figure 12, a given block of transform coefficients is first mapped on a 1-D array according to a predefined scanning pattern. Typically, after quantization a block contains only a few significant, i.e., nonzero coefficients, where, in addition, a predominant occurrence of coefficient levels with magnitude equal to 1, so-called trailing 1's (T1), is observed at the end of the scan. Therefore, as a preamble, first the number of nonzero coefficients and the number of T1s are transmitted using a combined codeword, where one out of four VLC tables are used based on the number of significant levels of neighboring blocks. Then, in the second step, sign and level value of significant coefficients are encoded by scanning the list of coefficients in revers order. By doing so, the VLC for coding each individual level value is adapted on the base of the previously encoded level by choosing among six VLC tables. Finally, the zero quantized coefficients are signaled by transmitting the total number of zeros before the last nonzero level for each block, and additionally, for each significant level the corresponding run, i.e., the number of consecutive preceding zeros. By monitoring the maximum possible number of zeros at each coding stage, a suitable VLC is chosen for the coding of each run value. A total number of 32 different VLCs are used in CAVLC entropy coding mode, where, however, the structure of some of these VLCs enables simple on-line calculation of any code word without recourse to the storage of code tables. For typical coding conditions and test material, bit rate

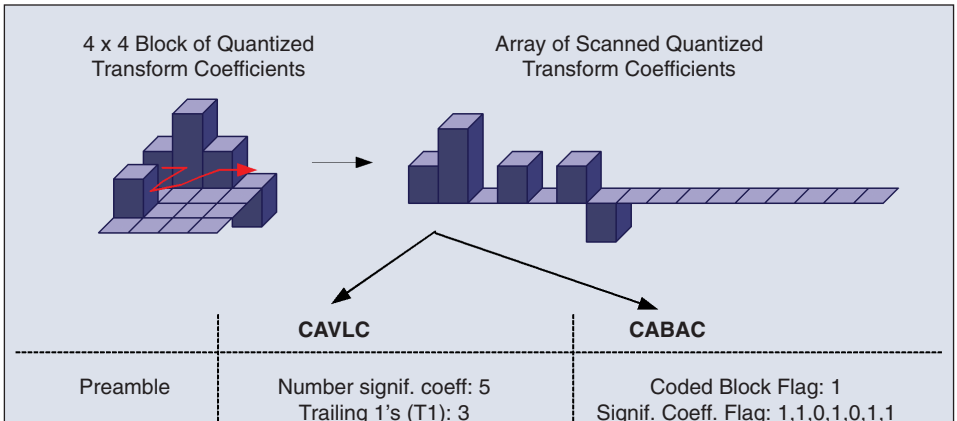


Figure 12. Precoding a block of quantized transform coefficients.

reductions of 2–7% are obtained by CAVLC relative to a conventional run-length scheme based on a single Exp-Golomb code.

For significantly improved coding efficiency, CABAC as the alternative entropy coding mode of H.264/AVC is the method of choice (Figure 13). As shown in Figure 13, the CABAC design is based on the key elements: binarization, context modeling, and binary arithmetic coding. Binarization enables efficient binary arithmetic coding via a unique mapping of non-binary syntax elements to a sequence of bits, a so-called bin string. Each element of this bin string can either be processed in the regular coding mode or the bypass mode. The latter is chosen for selected bins such as for the sign information or lower significant bins, in order to speedup the whole encoding (and decoding) process by means of a simplified coding engine bypass. The regular coding mode provides the actual coding benefit, where a bin may be context modeled and subsequently arithmetic encoded. As a design decision, in general only the most probable bin of a syntax element is supplied with a context model using previously encoded bins. Moreover, all regular encoded bins are adapted by estimating their actual probability distribution. The probability estimation and the actual binary arithmetic coding is conducted using a multiplication-free method that enables efficient implementations in hardware and software. Note that for coding of transform coefficients, CABAC is applied to specifically designed syntax elements, as shown in the example of Figure 12. Typically, CABAC provides bit rate reductions of 5–15% compared to CAVLC. More details on CABAC can be found in [16].

3.5 *Adaptive Deblocking Filter*

The block-based structure of the H.264/AVC architecture containing 4×4 transforms and block-based motion compensation, can be the source of severe blocking artifacts. Filtering the block edges has been shown to be a powerful tool to reduce the visibility of these artifacts. Deblocking can in principle be carried out as post-filtering, influencing only the pictures to be displayed. Higher visual quality can be achieved though, when the filtering process is carried out in the coding loop, because then all involved past reference frames used for motion compensation will be the filtered versions of the reconstructed frames. Another reason to make deblocking a mandatory in-loop tool in H.264/AVC is to enforce a decoder to approximately deliver a quality to the customer, which was intended by the producer and not leaving this basic picture enhancement tool to the optional good will of the decoder manufacturer.

The filter described in the H.264/AVC standard is highly adaptive. Several parameters and thresholds and also the local characteristics of the picture itself control the strength of the filtering process. All involved thresholds are quantizer dependent, because blocking artifacts will always become more severe when quantization gets coarse.

H.264/MPEG-4 AVC deblocking is adaptive on three levels:

- **On slice level,** the global filtering strength can be adjusted to the individual characteristics of the video sequence.
- **On block edge level,** the filtering strength is made dependent on inter/intra prediction decision, motion differences, and the presence of coded residuals in the two participating blocks. From these variables a filtering-strength parameter is calculated, which can take values from 0 to 4 causing modes from no filtering to very strong filtering of the involved block edge.
- **On sample level,** it is crucially important to be able to distinguish between true edges in the image and those created by the quantization of the transform-coefficients. True edges should be left unfiltered as much as possible. In order to separate the two cases, the sample values across every edge are analyzed. For an explanation denote the sample values inside two neighboring 4×4 blocks as $p_3, p_2, p_1, p_0 \mid q_0, q_1, q_2, q_3$ with the

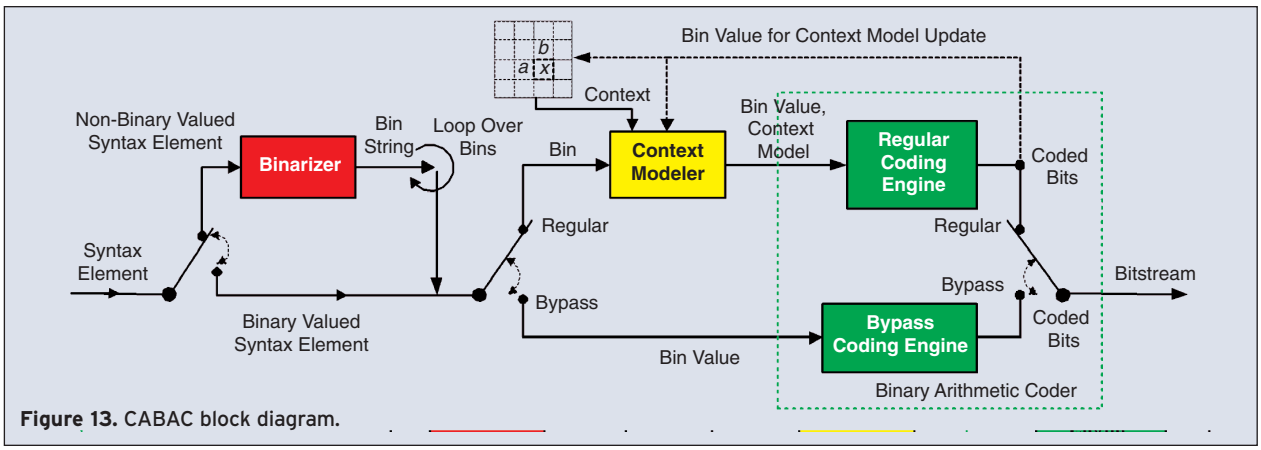


Figure 13. CABAC block diagram.

actual boundary between p_0 and q_0 as shown in Figure 14. Filtering of the two pixels p_0 and q_0 only takes place, if their absolute difference falls below a certain threshold α. At the same time, absolute pixel differences on each side of the edge ($|p_1 - p_0|$ *and* $|q_1 - q_0|$) have to fall below another threshold β, which is considerably smaller than α. To enable filtering of $p_1(q_1)$, additionally the absolute difference between p_0 and p_2 (q_0 and q_2) has to be smaller than β. The dependency of α and β on the quantizer, links the strength of filtering to the general quality of the reconstructed picture prior to filtering. For small quantizer values the thresholds both become zero, and filtering is effectively turned off altogether.

All filters can be calculated without multiplications or divisions to minimize the processor load involved in filtering. Only additions and shifts are needed. If filtering is turned on for p_0, the impulse response of the involved filter would in principle be (0, 1, 4, | 4, −1, 0) / 8. For p_1 it would be (4, 0, 2, | 2, 0, 0) / 8. The term in principle means that the maximum changes allowed for p_0 and p_1 (q_0 and q_1) are clipped to relatively small quantizer dependent values, reducing the low pass characteristic of the filter in a nonlinear manner.

Intra coding in H.264/AVC tends to use INTRA_16×16 prediction modes when coding nearly uniform image areas. This causes small amplitude blocking artifacts at the macro block boundaries which are perceived as abrupt steps in these cases. To compensate the resulting tiling artifacts, very strong low pass filtering is applied on boundaries between two macro blocks with smooth image content. This special filter also involves pixels p_3 and q_3.

In general deblocking results in bit rate savings of around 6–9% at medium qualities. More remarkable are the improvements in subjective picture quality. A more concise description of the H.264/AVC deblocking scheme can be found in [17].

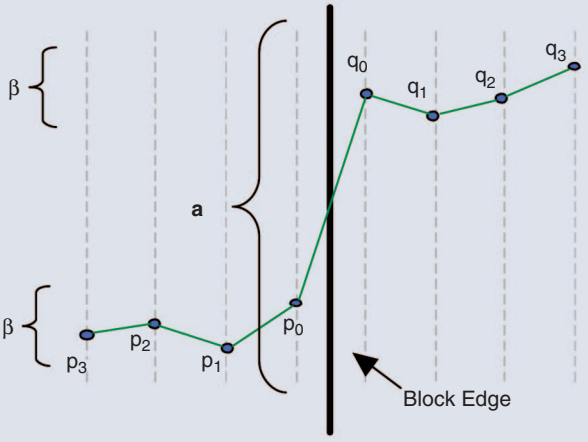


Figure 14. One-dimensional visualization of a block edge in a typical situation where the filter would be turned on.

3.6 *Error Robustness and Network Friendliness*

For efficient transmission in different environments, the seamless and easy integration of the coded video into all current and future protocol and network architectures is important. Therefore, both the VCL and the NAL are part of the H.264/AVC standard (Figure 2). The VCL specifies an efficient representation for the coded video signal. The NAL defines the interface between the video codec itself and the outside world. It operates on NAL *units* which give support to the packet-based approach of most existing networks. In addition to the NAL concept, the VCL itself includes several features providing network friendliness and error robustness being essential especially for real-time services such as streaming, multicasting, and conferencing applications due to online transmission and decoding. The H.264/AVC *Hypothetical Reference Decoder* (HRD) [18] places constraints on encoded NAL unit streams in order to enable cost-effective decoder implementations by introducing a multiple-leaky-bucket model.

Lossy and variable bit rate (VBR) channels such as the Internet or wireless links require channel-adaptive streaming or multi-casting technologies. Among others [19], channel-adaptive packet dependency control [20] and packet scheduling [21] allow reacting to these channels when transmitting pre-encoded video streams. These techniques are supported in H.264/AVC by various means, namely frame dropping of non-reference frames resulting in well-known temporal scalability, the *multiple reference frame concept* in combination with generalized B-pictures allowing a huge flexibility on frame dependencies to be exploited for temporal scalability and rate shaping of encoded video, and the possibility of switching between different bit streams which are encoded at different bit rates. This technique is called version switching. It can be applied at Instantaneous Decoder Refresh (IDR) frames, or, even more efficiently by the usage of switching pictures which allow identical reconstruction of frames even when different reference frames are being used. Thereby, *switching-predictive (SP) pictures* efficiently exploit motion-compensated prediction whereas *switching-intra (SI) pictures* can exactly reconstruct SP pictures. The switching between two bit streams using SI and SP pictures is illustrated in Figure 15 and Figure 16. Switching pictures can also be applied for error resilience purposes as well as other features, for details see [22].

Whereas for relaxed-delay applications such as download-and-play, streaming, and broadcast/multicast, residual errors can usually be avoided by applying powerful forward error correction and retransmission protocols, the low delay requirements for conversational applica-

tions impose additional challenges as transmission errors due to congestions and link-layer imperfectness can generally not be avoided. Therefore, these video applications require *error resilience features*. The H.264/AVC standardization process acknowledged thisby adopting a set of common test conditions for IP based transmission [23]. Anchor video sequences, appropriate bit rates and evaluation criteria are specified. In the following we briefly present different error resilience features included in the standard, for more details we refer to [24] and [7]. The presentation is accompanied by Figure 18 showing results for a representative selection of the common Internet test conditions, namely for the QCIF sequence Foreman 10 seconds are encoded at a frame rate of 7.5 fps applying only temporally backward referencing motion compensation. The resulting total bit rate including a 40 byte IP/UDP/RTP header matches exactly 64 kbit/s. As performance measure the average luminance peak signal to noise ratio (PSNR) is chosen and sufficient statistics are obtained by transmitting at least 10000 data packets for each experiment as well as applying a simple packet loss simulator and Internet error patterns[1] as specified in [23].

Although common understanding usually assumes that increased *compression efficiency* decreases error resilience, the opposite is the case if applied appropriately. As higher compression allows using additional bit rate for forward error correction, the loss probability of highly compressed data can be reduced assuming a constant overall bit rate. All other error resilience tools discussed in the following generally increase the data rate at the same quality, and, therefore, their application should always be considered very carefully in order not to effect adversely compression efficiency, especially if lower layer error protection is applicable. This can be seen for packet error rate 0 in Figure 18. *Slice structured coding* reduces packet loss probability and the visual degradation from packet losses, especially in combination with advanced decoder error concealment methods [25]. A slice is a sequence of macroblocks within one slice group and provides spatially distinct resynchronization points within the video data for a single frame. No intra-frame prediction takes place across slice boundaries. However,

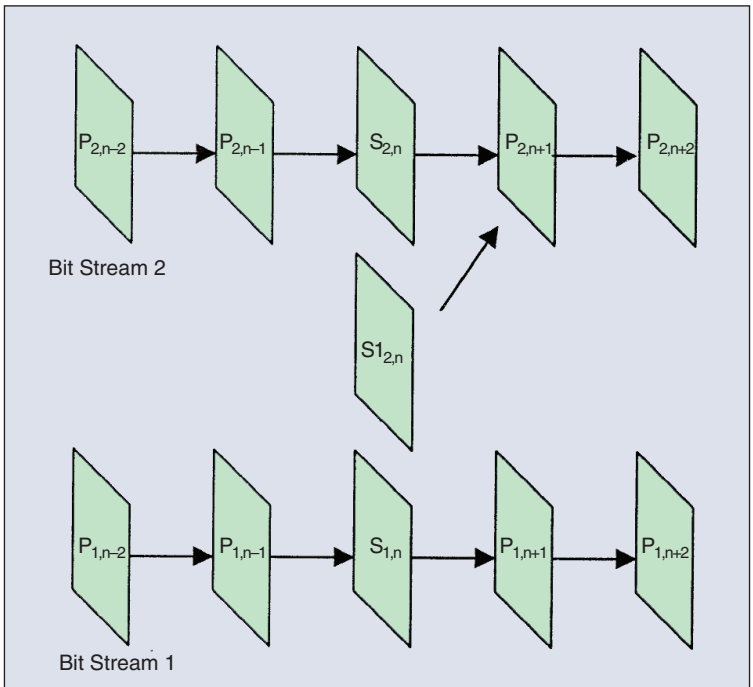


Figure 15. Switching between bit streams by using SI-Frames (from [22]).

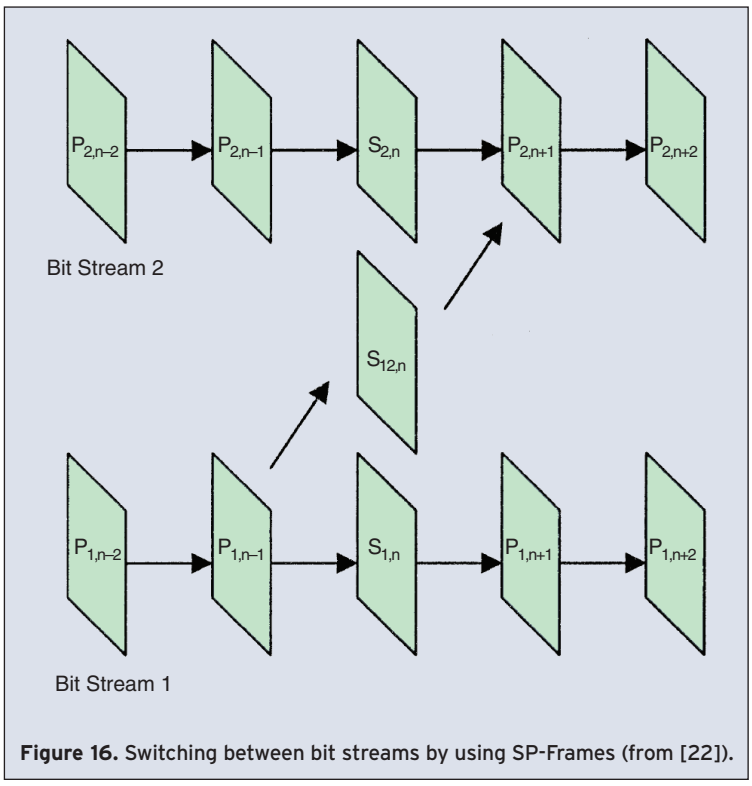


Figure 16. Switching between bit streams by using SP-Frames (from [22]).

[1]The Internet error pattern has been captured from real-world measurements and results in packet loss rates of approximately 3%, 5%, 10%, and 20%. These error probabilities label the packet error rate in Figure 18. Note that the 5% error file is burstier than the others resulting in somewhat unexpected results.

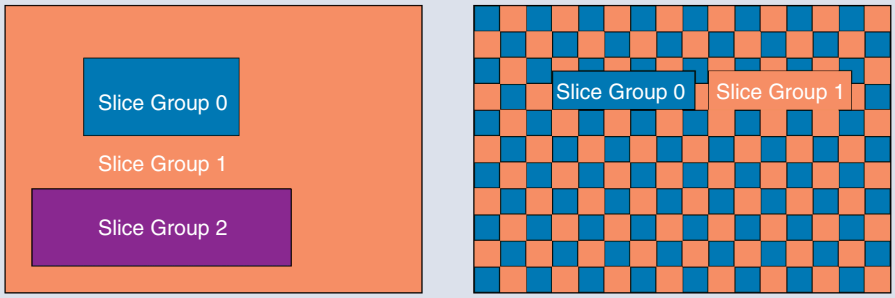


Figure 17. Division of an image into several slice groups using Flexible Macroblock Ordering (FMO).

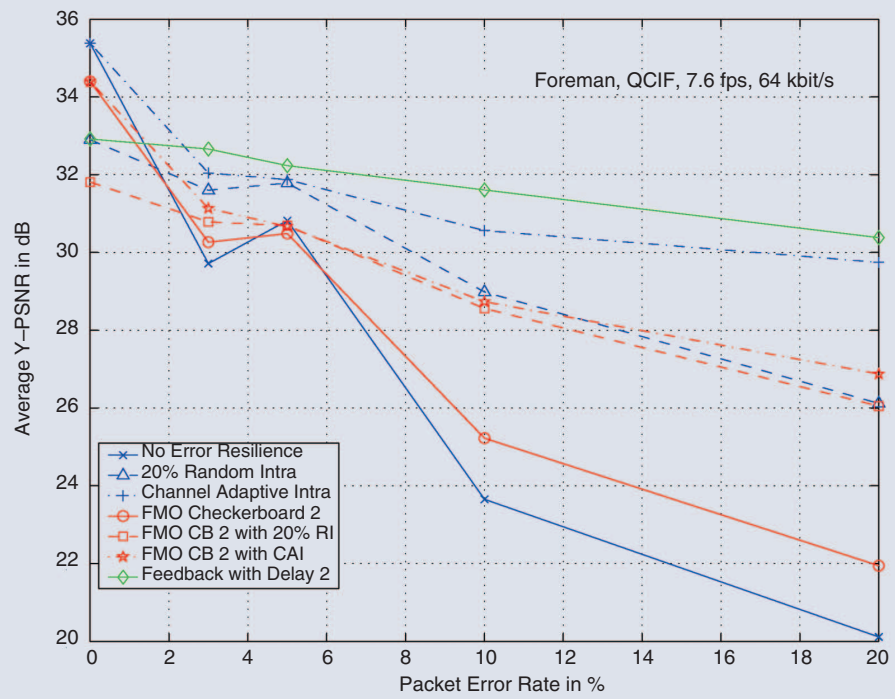


Figure 18. Average Y-PSNR over packet error rate (burstiness at 5% error rate is higher than for other error rates) for Foreman, QCIF, 7.5 fps, 64 kbit/s and different error resilience tools in H.264/AVC: no error resilience with one packet per frame, additional 20% random intra (RI) update, channel adaptive intra (CAI) update, each feature combined with FMO checkerboard pattern with 2 packets per frame (i.e., macroblocks with odd addresses in slice group 1, with even addresses in slice group 2), and feedback system with a 2-frame delayed (about 250 ms) decoder channel information at the encoder.

the loss of intra-frame prediction and the increased overhead associated with decreasing slice sizes adversely affect coding performance. Especially for wireless transmission a careful selection of the packet size is necessary [7].

As a more advanced feature, *Flexible Macroblock Ordering* (FMO) allows the specification of macroblock allocation maps defining the mapping of macroblocks to slice groups, where a slice group itself may contain several slices. An example is shown in Figure 17.

Therefore, macroblocks might be transmitted out of raster scan order in a flexible and efficient way. Specific macroblock allocation maps enable the efficient application of features such as slice interleaving, dispersed macroblock allocation using checkerboard-like patterns, one or several foreground slice groups and one left-over background slice groups, or sub-pictures within a picture to support, e.g., isolated regions [26]. Figure 18 shows increased performance for FMO with checkerboard pattern for increasing error rate when compared to the abandoning of error resilience features.

Arbitrary slice ordering (ASO) allows that the decoding order of slices within a picture may not follow the constraint that the address of the first macroblock within a slice is monotonically increasing within the NAL unit stream for a picture. This permits, for example, to reduce decoding delay in case of out-of-order delivery of NAL units.

Data Partitioning allows up to three partitions for the transmission of coded information. Rather than just providing two partitions, one for the header and the motion information, and one for the coded transform coefficients, H.264/AVC can generate three partitions by separating the second partition in intra and inter information. This allows assigning higher priority to, in general, more important intra information. Thus, it can reduce visual artifacts resulting from packet losses, especially if prioritization or unequal error protection is provided by the network.

If despite of all these techniques, packet losses and spa-

tio-temporal error propagation are not avoidable, quick recovery can only be achieved when image regions are encoded in Intra mode, i.e., without reference to a previously coded frame. H.264/AVC allows encoding of single macroblocks for regions that cannot be predicted efficiently. This feature can also be used to limit error propagation by transmitting a number of *intra coded macroblocks* anticipating transmission errors. The selection of Intra coded MBs can be done either randomly, in certain update patterns, or preferably in channel-adaptive rate-distortion optimized way [7], [27]. Figure 18 reveals that the introduction of intra coded macroblocks significantly improves the performance for increased error rates and can be combined with any aforementioned error resilience features. Thereby, channel-adaptive intra updates can provide better results than purely random intra updates, especially over the entire range of error rates.

A *redundant coded slice* is a coded slice that is a part of a *redundant picture* which itself is a coded representation of a picture that is not used in the decoding process if the corresponding primary coded picture is correctly decoded. Examples of applications and coding techniques utilizing the redundant coded picture feature include the video redundancy coding [28] and protection of "key pictures" in multicast streaming [29].

In bi-directional conversational applications it is common that the encoder has the knowledge of experienced NAL unit losses at the decoder, usually with a small delay. This small information can be conveyed from the decoder to the encoder. Although retransmissions are not feasible in a low-delay environment, this information is still useful at the encoder to limit error propagation [30]. The flexibility provided by the *multiple reference frame* concept in H.264/AVC allows incorporating so called NEWPRED approaches [31] in a straight-forward manner which address the problem of error propagation. For most successful applications, a selection of reference frames and intra updates can be integrated in a rate-distortion optimized encoder control as discussed in Section 4 taking into account not only video statistics, but also all available channel information [7]. Excellent results are shown in Figure 18 applying five reference frames and feedback delay of two frames, especially for moderate to higher error rates. To improve the performance also for low error rates, a combination of channel adaptive intra updates and feedback might be considered according to [27] at the expense of increased encoding complexity.

4. Rate Constrained Encoder Control

Due to the fact that the standard defines only the bitstream syntax and the possible coding tools the coding efficiency is dependent on the coding strategy of the encoder, which is not part of the standard (see Figure 1). Figure 19 shows the principle rate distortion working points for different encoder strategies. If just the minimization of the distortion is considered for the decision of the coding tools the achieved distortion is small but the required rate is very high. Vice versa, if just the rate is considered the achieved rate is small but the distortion is high. Usually, these working points are both not desired. Desired is a working point at which both the distortion and the rate are minimized together. This can be achieved by using Lagrangian optimization techniques, which are described for example in [32].

For the encoding of video sequences using the H.264/AVC standard, Lagrangian optimization techniques for the choice of the macroblock mode and the estimation of the displacement vector are proposed in [10], [33] and [34].

The macroblock mode of each macroblock S_k can be

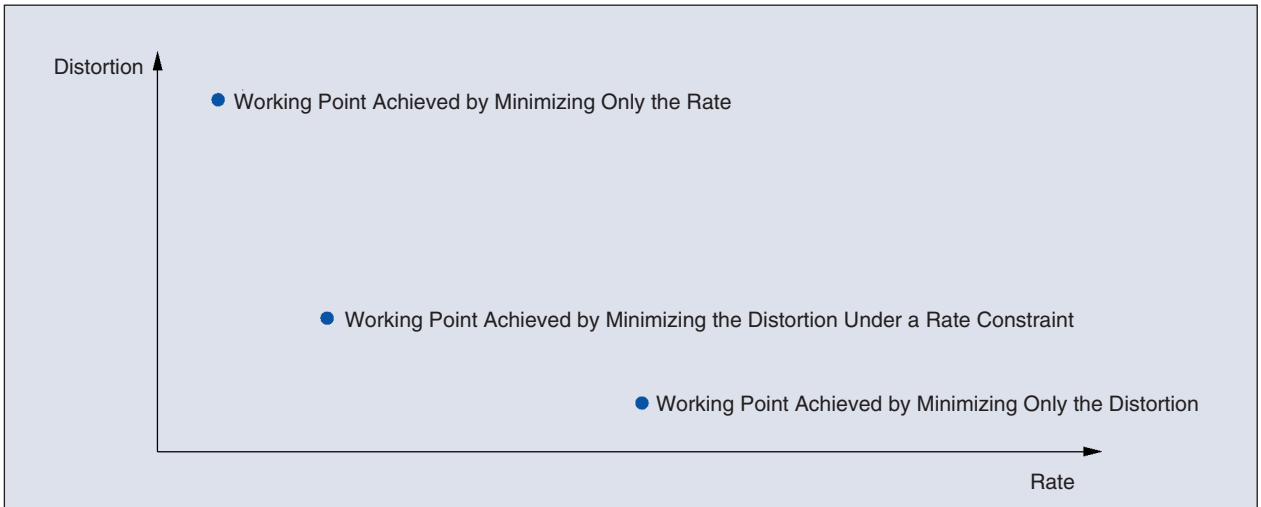


Figure 19. Principle rate distortion working points for different encoder strategies.

efficiently chosen out of all possible modes I_k by minimizing the functional

$$D_{\mathrm{REC}}(S_k, I_k \mid QP) + \lambda_{\mathrm{Mode}} \cdot R_{\mathrm{REC}}(S_k, I_k \mid QP) \rightarrow \min$$

Hereby the distortion D_{REC} is measured by the sum of squared differences (SSD) between the original signal s and the corresponding reconstructed signal s' of the same macroblock. The SSD can be calculated by

$$SSD = \sum_{(x,y)} \mid s[x, y, t] - s'[x, y, t] \mid^2.$$

The rate R_{REC} is the rate that is required to encode the block with the entropy coder. QP is the quantization parameter used to adjust the quantization step size. It ranges from 0 to 51.

The motion vectors can be efficiently estimated by minimizing the functional

$$D_{DFD}(S_i, \vec{d}) + \gamma_{\mathrm{Motion}} \cdot R_{\mathrm{Motion}}(S_i, \vec{d}) \rightarrow \min$$

with

$$D_{DFD}(S_i, \vec{d}) = \sum_{(x,y)} |s[x, y, t] - s'[x, -d_x, y, -d_y, t - d_t] |^2.$$

Hereby R_{Motion} is the rate required to transmit the motion information $\vec{d}$, which consists of both displacement vector components d_x and d_y and the corresponding reference frame number d_t. The following Lagrangian parameters lead to good results as shown in [10]:

$$\lambda_{\mathrm{Mode}} = \lambda_{\mathrm{Motion}} = 0.85.2^{(QP-12)/3}.$$

As already discussed, the tools for increased error resilience, in particular those to limit error propagation, do not significantly differ from those used for compression efficiency. Features like multi-frame prediction or macroblock intra coding are not exclusively error resilience tools. This means that bad decisions at the encoder can lead to poor results in coding efficiency or error resiliency or both. The selection of the coding mode for compression efficiency can be modified taking into account the influence of the random lossy channel. In this case, the encoding distortion is replaced by the expected decoder distortion. For the computation of the expected distortion we refer to, e.g. [27] or [35]. This method has been applied to generate channel-adaptive results in subsection 3.6 assuming a random-lossy channel with known error probability at the encoder.

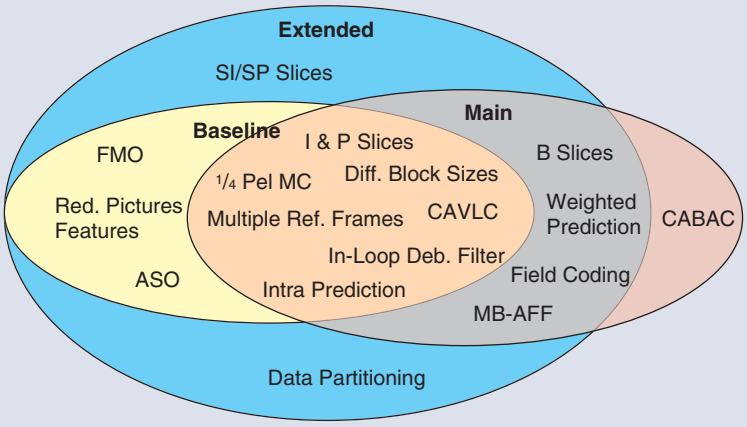


Figure 20. H.264/AVC profiles and corresponding tools.

5. Profiles and Levels of H.264/AVC

H.264/AVC has been developed to address a large range of applications, bit rates, resolutions, qualities, and services; in other words, H.264/AVC intends to be as generically applicable as possible. However, different applications typically have different requirements both in terms of functionalities, e.g., error resilience, compression efficiency and delay, as well as complexity (in this case, mainly decoding complexity since encoding is not standardized).

In order to maximize the interoperability while limiting the complexity, targeting the largest deployment of the standard, the H.264/AVC specification defines profiles and levels. A profile is defined as a subset of the entire bit stream syntax or in other terms as a subset of the coding tools. In order to achieve a subset of the complete syntax, flags, parameters, and other syntax elements are included in the bit stream that signal the presence or absence of syntactic elements that occur later in the bit stream. All decoders compliant to a certain profile must support all the tools in the corresponding profile.

However, within the boundaries imposed by the syntax of a given profile, there is still a large variation in terms of the capabilities required of the decoders depending on the values taken by some syntax elements in the bit stream such as the size of the decoded pictures. For many applications, it is currently neither practical nor economic to implement a decoder able to deal with all hypothetical uses of the syntax within a particular profile. To address this problem, a second profiling dimension was created for each profile: the levels. A level is a specified set of constraints imposed on values of the syntax elements in the bit stream.

These constraints may be simple limits on values or alternatively they may take the form of constraints on arithmetic combinations of values (e.g. picture width multiplied by picture height multiplied by number of pictures decoded per second) [1]. In H.264/AVC, the same level definitions are used for all profiles defined. However, if a certain terminal supports more than one profile, there is no obligation that the same level is supported for the various profiles. A profile and level combination specifies the so-called conformance points, this means points of interoperability for applications with similar functional requirements.

Summing up, profiles and levels together specify restrictions on the bit streams and thus minimum bounds on the decoding capabilities, making possible to implement decoders with different limited complexity, targeting different application domains. Encoders are not required to make use of any specific set of tools; they only have to produce bit streams which are compliant to the relevant profile and level combination.

To address the large range of applications considered by H.264/AVC, three profiles have been defined (see Figure 20):

- **Baseline Profile—**Typically considered the simplest profile, includes all the H.264/AVC tools with the exception of the following tools: B-slices, weighted prediction, field (interlaced) coding, picture/macroblock adaptive switching between frame and field coding (MB-AFF), CABAC, SP/SI slices and slice data partitioning. This profile typically targets applications with low complexity and low delay requirements.
- **Main Profile—**Supports together with the Baseline profile a core set of tools (see Figure 20); however, regarding Baseline, Main does exclude FMO, ASO and redundant pictures features while including B-slices, weighted prediction, field (interlaced) coding, picture/macroblock adaptive switching between frame and field coding (MB-AFF), and CABAC. This profile typically allows the best quality at the cost of higher complexity (essentially due to the B-slices and CABAC) and delay.
- **Extended Profile—**This profile is a superset of the Baseline profile supporting all tools in the specification with the exception of CABAC. The SP/SI slices and slice data partitioning tools are only included in this profile.

From Figure 20, it is clear that there is a set of tools supported by all profiles but the hierarchical capabilities for this set of profiles are reduced to Extended being a superset of Baseline. This means, for example, that only certain Baseline compliant streams may be decoded by a decoder compliant with the Main profile.

Although it is difficult to establish a strong relation between profiles and applications (and clearly nothing is normative in this regard), it is possible to say that conversational services will typically use the Baseline profile, entertainment services the Main profile, and streaming services the Baseline or Extended profiles for wireless or wired environments, respectively. However, a different approach may be adopted and, for sure, may change in time as additional complexity will become more acceptable.

In H.264/AVC, 15 levels are specified for each profile. Each level specifies upper bounds for the bit stream or lower bounds for the decoder capabilities, e.g., in terms of picture size (from QCIF to above 4k×2k), decoder processing rate (from 1485 to 983040 macroblocks per second), size of the memory for multi-picture buffers, video bit rate (from 64 kbit/s to 240 Mbit/s), and motion vector range (from [−64, +63.75] to [−512, +511.75]). For more detailed information on the H.264/AVC profiles and levels, refer to Annex A of [1].

6. Comparison to Previous Standards

In this section, a comparison of H.264/AVC to other video coding standards is given with respect to the coding efficiency (Subsection 6.1) and hardware complexity (Subsection 6.2).

6.1 *Coding Efficiency*

In [10], a detailed comparison of the coding efficiency of different video coding standards is given for video streaming, video conferencing, and entertainment-quality applications. All encoders are rate-distortion optimized using rate constrained encoder control [10], [33], [34]. For video streaming and video conferencing applications, we use test video sequences in the Common Intermediate Format (CIF, 352 × 288 picture elements, progressive) and in the Quarter Common Intermediate Format (QCIF, 176×144 picture elements, progressive). For entertainment-quality applications, sequences in ITU-R 601 (720×576 picture elements, interlaced) and High Definition Television (HDTV, 1280 × 720 picture elements, progressive) are used. The coding efficiency is measured by average bit rate savings for a constant peak signal to noise ratio (PSNR). Therefore the required bit rates of several test sequences and different qualities are taken into account.

For video streaming applications, H.264/AVC MP (Main Profile), MPEG-4 Visual ASP (Advanced Simple Profile), H.263 HLP (High Latency Profile), and MPEG-2 Video ML@MP (Main Level at Main Profile) are considered. Figure 21 shows the PSNR of the luminance component versus the average bit rate for the single test sequence *Tempete* encoded at 15 Hz and Table 1 presents the average bit rate savings for a variety of test sequences and bit

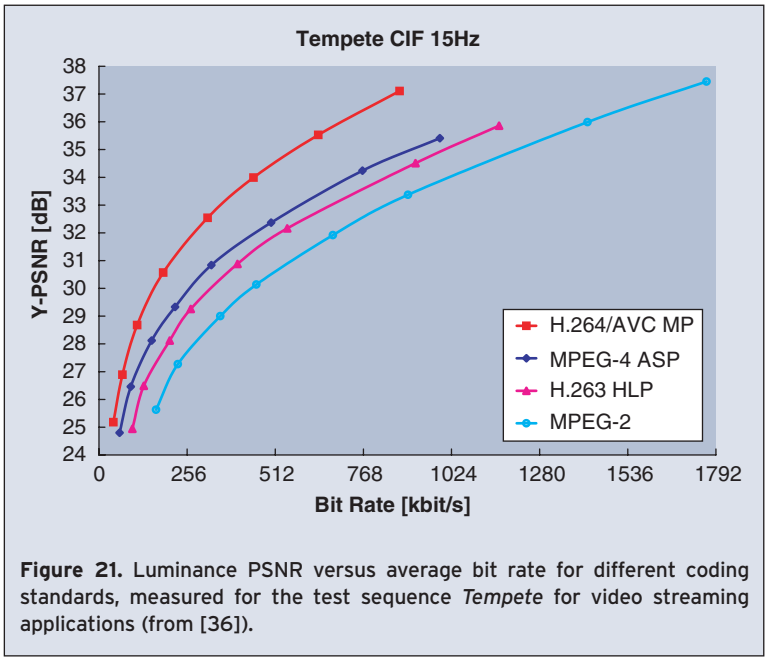


Figure 21. Luminance PSNR versus average bit rate for different coding standards, measured for the test sequence *Tempete* for video streaming applications (from [36]).

rates. It can be drawn from Table 1 that H.264/AVC outperforms all other considered encoders. For example, H.264/AVC MP allows an average bit rate saving of about 63% compared to MPEG-2 Video and about 37% compared to MPEG-4 Visual ASP.

For video conferencing applications, H.264/AVC BP (Baseline Profile), MPEG-4 Visual SP (Simple Profile), H.263 Baseline, and H.263 CHC (Conversational High Compression) are considered. Figure 22 shows the luminance PSNR versus average bit rate for the single test sequence *Paris* encoded at 15 Hz and Table 2 presents the average bit rate savings for a variety of test sequences and bit rates. As for video streaming applications, H.264/AVC outperforms all other considered encoders. H.264/AVC BP allows an average bit rate saving of about 40% compared to H.263 Baseline and about 27% compared to H.263 CHC.

For entertainment-quality applications, the average bit rate saving of H.264/AVC compared to MPEG-2 Video ML@MP and HL@MP is 45% on average [10]. A part of this gain in coding efficiency is due to the fact that H.264/AVC achieves a large degree of removal of film grain noise resulting from the motion picture production process. However, since the perception of this noisy grain texture is often considered to be desirable, the difference in perceived quality between H.264/AVC coded video and MPEG-2 coded video may often be less distinct than indicated by the PSNR-based comparisons, especially in high-quality, high-resolution applications such as High-Definition DVD or Digital Cinema.

In certain applications like the professional motion picture production, random access for each individual picture may be required. Motion-JPEG2000 [37] as an extension of the new still image coding standard JPEG2000 provides this feature along with some useful scalability properties. When restricted to IDR frames, H.264/AVC is also capable of serving the needs for such a random access capability. Figure 23 shows PSNR for the luminance component versus average bit rate for the ITU-R 601 test sequence *Canoe* encoded in intra mode only, i.e., each field of the whole sequence is coded in intra mode only. Interestingly, the measured rate-distortion performance of H.264/AVC MP is better than that of the state-of-the-art in still image compression as exemplified by JPEG2000, at least in this particular test case. Other test cases were studied in [38] as well, leading to a general observation that up to 1280 × 720 pel HDTV signals the pure intra coding performance of H.264/AVC MP is comparable or better than that of Motion-JPEG2000.

Table 1.
Average bit rate savings for video streaming applications (from [10]).

Coder	Average Bit Rate Savings Relative To: MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC MP	37.44%	47.58%	63.57%
MPEG-4 ASP	–	16.65%	42.95%
H.263 HLP	–	–	30.61%

Table 2.
Average bit rate savings for video conferencing applications (from [10]).

Coder	Average Bit Rate Savings Relative To: H.263 CHC	MPEG-4 SP	H.263 Base
H.264/AVC BP	27.69%	29.37%	40.59%
H.263 CHC	–	2.04%	17.63%
MPEG-4 SP	–	–	15.69%

6.2 *Hardware Complexity*

Assessing the complexity of a new video coding standard is not a straightforward task: its implementation complexity heavily depends on the characteristics of the platform (e.g., DSP processor, FPGA, ASIC) on which it is mapped. In this section, the data transfer characteristics are chosen as generic, platform independent, metrics to express implementation complexity. This approach is motivated by the data dominance of multimedia applications [39]–[44].

Both the size and the complexity of the specification and the intricate interdependencies between different H.264/AVC functionalities, make complexity assessment using only the paper specification unfeasible. Hence the

presented complexity analysis has been performed on the executable C code produced by the JVT instead. As this specification is the result of a collaborative effort, the code unavoidably has different properties with respect to optimisation and platform dependence. Still, it is our experience that when using automated profiling tools yielding detailed data transfer characteristics (such as [45]) on similar specifications (e.g., MPEG-4) meaningful relative complexity figures are obtained (this is also the conclusion of [46]. The H.264/AVC JM2.1 code is used for the reported complexity assessment experiments. Newer versions of the executable H.264/AVC specification have become available that also include updated tool definitions achieving a reduced complexity.

The test sequences used in the complexity assessment are: Mother & Daughter 30 Hz QCIF, Foreman 25 Hz QCIF and CIF, and Mobile & Calendar 15 Hz CIF (with bit rates ranging from 40 Kbits/s for the simple sequences to 2 Mbits/s for the complex ones). A fixed quantization parameter setting has been assumed.

The next two subsections highlight the main contributions to the H.264/AVC complexity. Consequently some general considerations are presented.

6.2.1 Complexity Analysis of Some Major H.264/AVC Encoding Tools

- *Variable Block Sizes*: using variable block sizes affects the access frequency in a linear way: more than 2.5% complexity increase[2] for each additional mode. A typical bit rate reduction between 4 and 20% is achieved (for the same quality) using this tool, however, the complexity increases linearly with the number of modes used, while the corresponding compression gain saturates.
- *Hadamard transform*: the use of Hadamard coding results in an increase of the access frequency of roughly 20%, while not significantly impacting the quality vs. bit rate for the test sequences considered.
- *RD-Lagrangian optimisation*: this tool comes with a data transfer increase in the order of 120% and improves PSNR (up to 0.35 dB) and bit rate (up to 9% bit savings). The performance vs. cost trade-off when using RD techniques for motion estimation and coding mode decisions inherently depends on the other tools used. For instance, when applied to a basic configuration with 1 reference frame and only 16×16 block size, the resulting complexity increase is less than 40%.
- *B-frames*: the influence of B frames on the access frequency varies from −16 to +12% depending on the test case and decreases the bit rate up to 10%.
- *CABAC*: CABAC entails an access frequency increase from 25 to 30%, compared to methods using a single reversible VLC table for all syntax elements,. Using CABAC reduces the bit rate up to 16%.

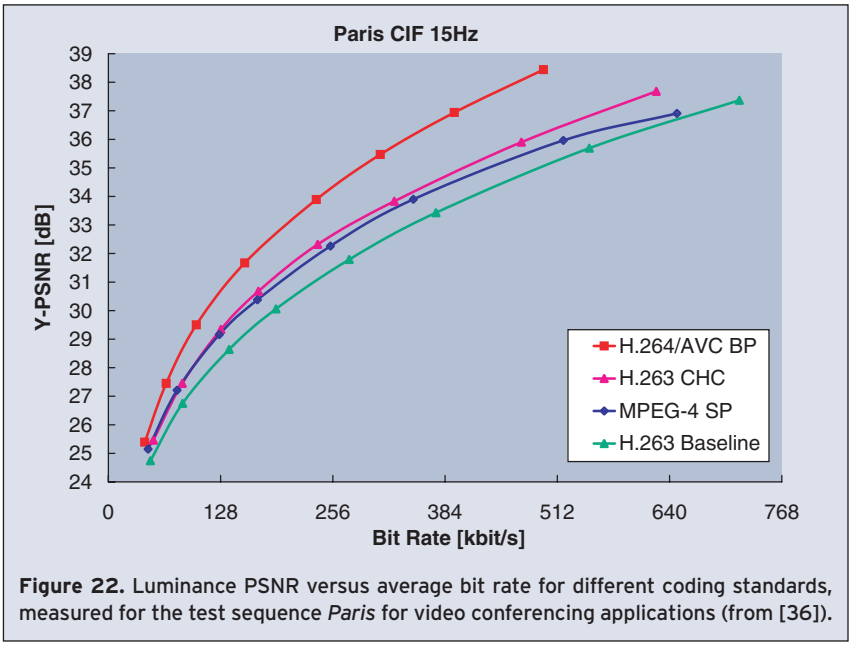


Figure 22. Luminance PSNR versus average bit rate for different coding standards, measured for the test sequence *Paris* for video conferencing applications (from [36]).

- *Displacement vector resolution*: The encoder may choose to search for motion vectors only at $^1/_2$ pel positions instead of $^1/_4$ pel positions. This results in a decrease of access frequency and processing time of about 10%. However, use of $^1/_4$ pel motion vectors increases coding efficiency up to 30% except for very low bit rates.
- *Search Range*: increasing both reference frame numbers and search size leads to higher access frequency, up to approximately 60 times (see also Table 3), while it has a minimal impact on PSNR and bit rate performances.
- *Multiple Reference Frames*: adopting multiple reference frames increases the access frequency accord-

[2]Complexity increases and compression improvements are relative to a comparable, meaningful configuration without the tool under consideration, see also [47].

Table 3.
Impact of the number of reference frames and search range on the number of encoder accesses (relative to the simplest case considered for each sequence).

	Foreman 25 Hz QCIF			Foreman 25 Hz CIF			Mobile & Calendar 15 Hz CIF		
Search Range	8	16	32	8	16	32	8	16	32
5 ref. frames	16.9	24.6	55.7	17.5	25.3	56.1	16.6	23.1	48.8
1 ref. frame	1	2.54	8.87	1	2.53	8.90	1	2.49	8.49

ing to a linear model: 25% complexity increase for each added frame. A negligible gain (less than 2%) in bit rate is observed for low and medium bit rates, but more significant savings can be achieved for high bit rate sequences (up to 14%).

- *Deblocking filter*: The mandatory use of the deblocking filter has no measurable impact on the encoder complexity. However, the filter provides a significant increase in subjective picture quality.

For the encoder, the main bottleneck is the combination of multiple reference frames and large search sizes. Speed measurements on a Pentium IV platform at 1.7 GHz with Windows 2000 are consistent with the above conclusions (this platform is also used for the speed measurements for the decoder).

6.2.2 Complexity Analysis of Some Major H.264/AVC Decoding Tools

- *CABAC*: the access frequency increase due to CABAC is up to 12%, compared to methods using a single reversible VLC table for all syntax elements,. The higher the bit rate, the higher the increase.
- *RD-Lagrangian optimization*: the use of Lagrangian cost functions at the encoder causes an average complexity increase of 5% at the decoder for middle and low–rates while higher rate video is not affected (i.e. in this case, encoding choices result in a complexity increase at the decoder side).
- *B-frames*: the influence of B-frames on the data transfer complexity increase varies depending on the test case from 11 to 29%. The use of B-frames has an important effect on the decoding time: introducing a first B-frame requires an extra 50% cost for the very low bit rate video, 20 to 35% for medium and high bite-rate video. The extra time required by the second B-frame is much lower (a few %).
- *Hadamard transform*: the influence on the decoder of using the Hadamard transform at the encoder is negligible in terms of memory accesses, while it increases the decoding time up to 5%.
- *Deblocking filter*: The use of the mandatory deblocking filter increases the decoder access frequency by 6%.
- *Displacement vector* resolution: In case the encoder sends only vectors pointing to $^1/_2$ pel positions, the access frequency and decoding time decrease about 15%.

6.2.3 Other Considerations

In relative terms, the encoder complexity increases with more than one order of magnitude between MPEG-4 Part 2 (Simple Profile) and H.264/AVC (Main Profile) and with a factor of 2 for the decoder. The H.264/AVC encoder/decoder complexity ratio is in the order of 10 for basic configurations and can grow up to 2 orders of magnitude for complex ones, see also [47].

Our experiments have shown that, when combining the new coding features, the relevant implementation

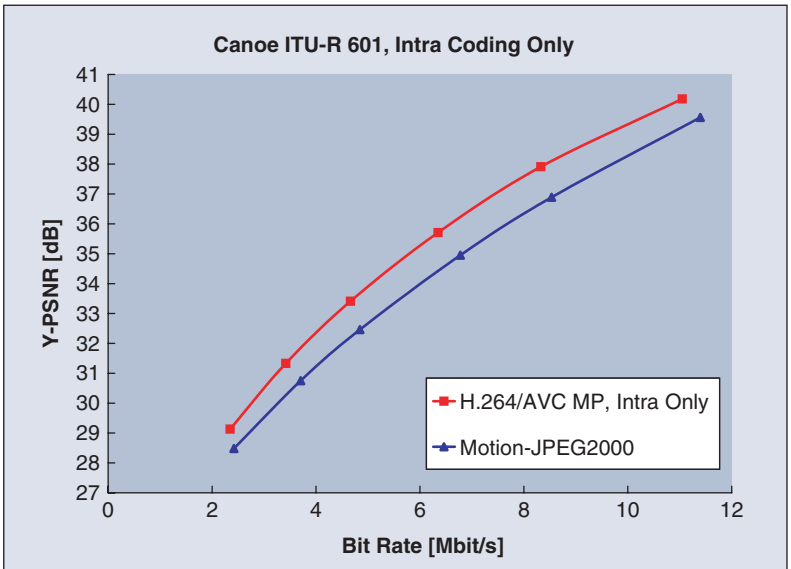


Figure 23. Luminance PSNR versus average bit rate for H.264/AVC and Motion-JPEG2000, measured for the ITU-R 601 test sequence *Canoe* for pure intra coding.

complexity accumulates while the global compression efficiency saturates. An appropriate use of the H.264/AVC tools leads to roughly the same compression performance if all the tools would be used simultaneously, but with a considerable reduction in implementation complexity (a factor 6.5 for the encoder and up to 1.5 for the decoder). These efficient use modes are reflected in the choice of the tools and parameter settings of the H.264/AVC profiles (see Section 5). More information on complexity analyses that have been performed in the course of H.264/AVC standardisation can be found in [48] [49] [50].

7. Licensing of H.264/AVC Technology

Companies and universities introducing technology into international standards usually protect their intellectual property with patents. When participants in the standards definition process proposed patented technology to be included into the standard they promised to license the use of their technology in fair, reasonable and non-discriminatory terms, the so-called RAND conditions. Essential patents describe technology that has to be implemented in a standards-compliant decoder. The use of patented technology requires a user of this technology to license it from the respective owner. Given that there are many patents used in any modern video coding standard, several companies pooled their patents into a pool such that licensing H.264/AVC technology is easy for the user. At this point, there are two patent pools: One is organized by MPEG LA and the other by Via Licensing. Since the patents covered by the two patent pools are not precisely the same, users of H.264/AVC technology need in principle to have a license from both patent pools. Unfortunately, these pools do not guarantee that they cover the entire technology of H.264 as participation of a patent owner in a patent pool is voluntary.

MPEG LA LLC is the organization which gathered the owners of essential patents like Columbia University, Electronics and Telecommunications Research Institute of Korea (ETRI), France Télécom, Fujitsu, LG Electronics, Matsushita, Mitsubishi, Microsoft, Motorola, Nokia, Phillips, Robert Bosch GmbH, Samsung, Sharp, Sony, Toshiba, and Victor Company of Japan (JVC) into a patent pool. VIA Licensing Corporation, a subsidiary of Dolby Laboratories, licenses essential H.264/AVC technology from companies like Apple Computer, Dolby Laboratories, FastVDO, Fraunhofer-Gesellschaft eV, IBM, LSI Logic, Microsoft, Motorola, Polycom, and RealNetworks. Both patent pools may be licensed for the commercial use of an H.264/AVC decoder. Unfortunately, the terms of the license differ.

MPEG LA terms: After the end of a grace period in December 2004, an end product manufacturer for encoders or decoders has to pay a unit fee of $0.20 per unit after the first 100,000 units that are free each year. In addition to this fee for the actual soft- or hardware, certain companies are taxed a participation fee starting January 2006. Providers of Pay-per-View, download or Video-on-Demand services pay the lower of 2% of the sales price or $0.02 for each title. This applies to all transmission media like cable, satellite, Internet, mobile and over the air. Subscription services with more than 100,000 but less than 1,000,000 AVC video subscribers pay a minimum of $0.075 and a maximum of $0.25 per subscriber per year. Operators of over-the-air free broadcast services are charged $10,000 per year per transmitter. Free Internet broadcast is exempt from any fees until December 2010.

VIA Licensing terms: After the end of a grace period in December 2004, an end product manufacturer for encoder or decoders has to pay a unit fee of $0.25 per unit. A participation or replication fee is not required if the content is provided for free to the users. A fee of $0.005 for titles shorter than 30 minutes up to $0.025 for titles longer than 90 minutes has to be paid for titles that are permanently sold. For titles that are sold on a temporary basis, the 'replication fee' is $0.0025. This patent pool does not require the payment of any fees as long as a company distributes less than 50,000 devices and derives less than $500,000 revenue from its activities related to devices and content distribution. It appears that interactive communication services like video telephony only requires a unit fee but not a participation fee. While previous standards like MPEG-2 Video also required a license fee to be paid for every encoder and decoder, the participation fees established for the use of H.264/AVC require extra efforts from potential commercial users of H.264/AVC.

Disclaimer: No reliance may be placed on this section on licensing of H.264/AVC technology without written confirmation of its contents from an authorized representative.

8. Summary

This new international video coding standard has been jointly developed and approved by the MPEG group ISO/IEC and the VCEG group of ITU-T. Compared to previous video coding standards, H.264/AVC provides an improved coding efficiency and a significant improvement in flexibility for effective use over a wide range of networks. While H.264/AVC still uses the concept of block-based motion compensation, it provides some significant changes:

- Enhanced motion compensation capability using high precision and multiple reference frames
- Use of an integer DCT-like transform instead of the DCT

- Enhanced adaptive entropy coding including arithmetic coding
- Adaptive in-loop deblocking filter

The coding tools of H.264/AVC when used in an optimized mode allow for bit savings of about 50% compared to previous video coding standards like MPEG-4 and MPEG-2 for a wide range of bit rates and resolutions. However, these savings come at the price of an increased complexity. The decoder is about 2 times as complex as an MPEG-4 Visual decoder for the Simple profile, and the encoder is about 10 times as complex as a corresponding MPEG-4 Visual encoder for the Simple profile. The H.264/AVC main profile decoder suitable for entertainment applications is about four times more complex than MPEG-2. The encoder complexity depends largely on the algorithms for motion estimation as well as for the rate-constrained encoder control. Given the performance increase of VLSI circuits since the introduction of MPEG-2, H.264/AVC today is less complex than MPEG-2 in 1994. At this point commercial companies may already license some technology for implementing an H.264/AVC decoder from two licensing authorities simplifying the process of building products on H.264/AVC technology.

9. Acknowledgments

The authors would like to thank the experts of ISO/IEC MPEG, ITU-T VCEG, and ITU-T/ISO/IEC Joint Video Team for their contributions in developing the standard.

10. References

[1] ISO/IEC 14496–10:2003, "Coding of Audiovisual Objects—Part 10: Advanced Video Coding," 2003, also ITU-T Recommendation H.264 "Advanced video coding for generic audiovisual services."
[2] ITU-T Recommendation H.261, "Video codec for Audiovisual Services at p X 64 kbit/s," March 1993.
[3] ISO/IEC 11172: "Information technology—coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," Geneva, 1993.
[4] ISO/IEC 13818–2: "Generic coding of moving pictures and associated audio information—Part 2: Video," 1994, also ITU-T Recommendation H.262.
[5] ITU-T Recommendation H.263, "Video Coding for Low bit rate Communication," version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.
[6] ISO/IEC 14496–2: "Information technology—coding of audiovisual objects—part 2: visual," Geneva, 2000.
[7] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems*, vol. 13, no. 7, pp. 657–673, July 2003.
[8] T. Wedi and H.G. Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 577–587, July 2003.
[9] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 70–84, Feb. 1999.
[10] T. Wiegand, H. Schwarz, A. Joch, and F. Kossentini, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 688–703, July 2003.
[11] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Processing*, vol. 9, Feb. 1999.
[12] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multi-hypothesis motion-compensated prediction for video coding," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Sept. 2000, vol. 3, pp. 150–153.
[13] N. Ahmed, T. Natarajan, and R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, Jan. 1974.
[14] Iain E G Richardson, "H.264/MPEG-4 Part 10 White Paper." Available: http://www.vcodex.fsnet.co.uk/resources.html
[15] H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-Complexity transform and quantization in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 598–603, July 2003.
[16] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620–636, July 2003.
[17] P. List, A. Joch, J. Lainema, and G. Bjontegaard, "Adaptive deblocking filter" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 614–619, July 2003.
[18] J. Ribas-Corbera, P.A. Chou, and S. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," *IEEE Transactions on Circuits and Systems*, vol. 13, no. 7, pp. 674–687, July 2003.
[19] B. Girod, M. Kalman, Y.J. Liang, and R. Zhang, "Advances in video channel-adaptive streaming," in *Proc. ICIP 2002*, Rochester, NY, Sept. 2002.
[20] Y.J. Liang and B. Girod, "Rate-distortion optimized low–latency video streaming using channel-adaptive bitstream assembly," in *Proc. ICME 2002*, Lausanne, Switzerland, Aug. 2002.
[21] S.H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Proc. International Packet Video Workshop 2002*, Pittsburgh, PY, April 2002.
[22] M. Karczewicz and R. Kurçeren, "The SP and SI frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems*, vol. 13, no. 7, pp. 637–644, July 2003.
[23] S. Wenger. (September 2001). Common Conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing. VCEG-N79r1. Available: http://standard.pictel. com/ftp/video-site/0109_San/VCEG-N79r1.doc,.
[24] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems*, vol. 13, no. 7, pp. 645–656, July 2003.
[25] Y.-K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. ICIP*, vol. 2, pp. 729–732, Sept. 2002.
[26] Y.-K. Wang, M.M. Hannuksela, and M. Gabbouj, "Error-robust inter/intra mode selection using isolated regions," in *Proc. Int. Packet Video Workshop 2003*, Apr. 2003.
[27] R. Zhang, S.L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE JSAC*, vol. 18, no. 6, pp. 966–976, July 2000.
[28] S. Wenger, "Video redundancy coding in H.263+," *1997 International Workshop on Audio-Visual Services over Packet Networks*, Sept. 1997.
[29] Y.-K. Wang, M.M. Hannuksela, and M. Gabbouj, "Error resilient video coding using unequally protected key pictures," in *Proc. International Workshop VLBV03*, Sept. 2003.
[30] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," in *Proc. of IEEE*, vol. 97, no. 10, Oct. 1999, pp. 1707–1723.
[31] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in *Proc. IEEE Globecom*, vol. 3, Nov. 1996.
[32] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15 no. 6, pp. 23–50, Nov. 1998.
[33] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, Nov. 1998.
[34] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," in *Proc. of ICIP 2001*, Thessaloniki, Greece, Oct. 2001.
[35] T. Stockhammer, D. Kontopodis, and T. Wiegand, "Rate-distortion optimization for H.26L video coding in packet loss environment," in *Proc. Packet Video Workshop 2002*, Pittsburgh, PY, April 2002.
[36] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," *Proc. of the IEEE ICIP 2002*, part II, pp. 501–504, Sept. 2002.
[37] ISO/IEC 15444-3, "Motion-JPEG2000" (JPEG2000 Part 3), Geneva, 2002.
[38] D. Marpe, V. George, H.L. Cycon, and K.U. Barthel, "Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in intra coding mode," in *Proc. SPIE Conf. on Wavelet Applications in Industrial Processing, Photonics East*, Rhode Island, USA, Oct. 2003.
[39] F. Catthoor, et al., *Custom Memory Management Methodology*.

Kluwer Academic Publishers, 1998
[40] J. Bormans et al., "Integrating system-level low power methodologies into a real-life design flow," in *Proc. IEEE PATMOS '99*, Kos, Greece, Oct. 1999, pp. 19–28.
[41] Chimienti, L. Fanucci, R. Locatelli, and S. Saponara, "VLSI architecture for a low-power video codec system," *Microelectronics Journal*, vol. 33, no. 5, pp. 417–427, 2002.
[42] T. Meng et al., "Portable video-on-demand in wireless communication," *Proc. of the IEEE*, vol. 83 no. 4, pp. 659–680, 1995.
[43] J. Jung, E. Lesellier, Y. Le Maguet, C. Miro, and J. Gobert, "Philips deblocking solution (PDS), a low complexity deblocking for JVT," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-B037, Geneva, CH, Feb. 2002.
[44] L. Nachtergaele et al., "System-Level power optimization of video codecs on embedded cores: A systematic approach," *Journal of VLSI Signal Processing*, Kluwer Academic Publisher, vol. 18, no. 2, pp. 89–111, 1998.
[45] http://www.imec.be/atomium
[46] S. Bauer, et al., "The MPEG-4 multimedia coding standard: Algorithms, architectures and applications," *Journal of VLSI Signal Processing*. Boston: Kluwer, vol. 23, no. 1, pp. 7–26, Oct. 1999.
[47] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "The JVT advanced video coding standard: Complexity and performance analysis on a tool-by-tool basis," *Packet Video Workshop (PV'03)*, Nantes, France, April 2003.
[48] V. Lappalainen et al., "Optimization of emerging H.26L video encoder," in *Proc. IEEE SIPS'01*, Sept. 2001, pp. 406–415.
[49] V. Lappalainen, A. Hallapuro, and T. Hamalainen, "Performance analysis of low bit rate H.26L video encoder," in *Proc. IEEE ICASSP'01*, May 2001, pp. 1129–1132.
[50] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Tran. Circ. Sys. Video Tech.*, vol. 13, no 7, pp. 715–727, 2003.

Jörn Ostermann studied Electrical Engineering and Communications Engineering at the University of Hannover and Imperial College London, respectively. He received Dipl.-Ing. and Dr.-Ing. from the University of Hannover in 1988 and 1994, respectively. From 1988 till 1994, he worked as a Research Assistant at the Institut für Theoretische Nachrichtentechnik conducting research in low bit-rate and object-based analysis-synthesis video coding. In 1994 and 1995 he worked in the Visual Communications Research Department at AT&T Bell Labs on video coding. He was a member of Image Processing and Technology Research within AT&T Labs—Research from 1996 to 2003. Since 2003 he is Full Professor and Head of the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung at the Universität Hannover, Germany.

From 1993 to 1994, he chaired the European COST 211 sim group coordinating research in low bitrate video coding. Within MPEG-4, he organized the evaluation of video tools to start defining the standard. He chaired the Adhoc Group on Coding of Arbitrarily-shaped Objects in MPEG-4 Video. Jörn was a scholar of the German National Foundation. In 1998, he received the AT&T Standards Recognition Award and the ISO award. He is a member of IEEE, the IEEE Technical Committee on Multimedia Signal Processing, past chair of the IEEE CAS Visual Signal Processing and Communications (VSPC) Technical Committee and a Distinguished Lecturer of the IEEE CAS Society. He published more than 50 research papers and book chapters. He is coauthor of a graduate level text book on video communications. He holds 10 patents. His current research interests are video coding and streaming, 3D modeling, face animation, and computer-human interfaces.

Matthias Narroschke was born in Hanover in 1974. He received his Dipl.-Ing. degree in electrical engineering from the University of Hanover in 2001 (with highest honors). Since then he has been working toward the PhD degree at the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung of the University of Hanover. His research interests include video coding, 3D image processing and video processing, and internet streaming. In 2003, he became a Senior Engineer. He received the Robert-Bosch-Prize for the best Dipl.-Ing. degree in electrical engineering in 2001. He is an active delegate to the Motion Picture Experts Group (MPEG).

Thomas Wedi received his Dipl.-Ing. degree in 1999 from the University of Hannover, Hannover, Germany, where he is currently working toward the Ph.D. degree with research focused on motion- and aliasing-compensated prediction for hybrid video coding.

He has been with the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, as Research Scientist and Teaching Assistant. In 2001 he also became the Senior Engineer. His further research interests include video coding and transmission, 3D image and video processing, and audio-visual communications. He is an active contributor to the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC/ITU-T Joint Video Team (JVT), where the H.264/AVC video coding standard is developed. In both standardization groups he chaired an Ad-Hoc group on interpolation filtering. In cooperation with Robert Bosch GmbH, he holds several international patents in the area of video compression.

Thomas Stockhammer received his Diplom-Ingenieur degree in electrical engineering from the Munich University of Technology (TUM), Germany, in 1996. Since then he has been working toward the Dr.-Ing. degree at the Munich University of Technology, Germany, in the area of multimedia and video transmission over mobile and packet-lossy channels. In 1996, he visited Rensselear Polytechnic Institute (RPI), Troy, NY, to perform his diploma thesis in the area of combined source-channel coding for video

and coding theory. There he started the research in video transmission as well as source and channel coding.

In 2000, he was Visiting Researcher in the Information Coding Laboratory at the University of San Diego, California (UCSD). Since then he has published numerous conference and journal papers and holds several patents. He regularly participates and contributes to different standardization activities, e.g. ITU-T H.324, H.264, ISO/IEC MPEG, JVT, IETF, and 3GPP. He acts as a member of several technical program committees, as a reviewer for different journals, and as an evaluator for the European Commission. His research interests include joint source and channel coding, video transmission, multimedia networks, system design, rate-distortion optimization, information theory, as well as mobile communications.

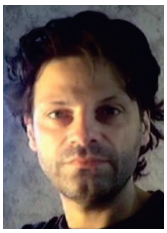
Jan Bormans, Ph.D., has been a researcher at the Information Retrieval and Interpretation Sciences laboratory of the Vrije Universiteit Brussel (VUB), Belgium, in 1992 and 1993. In 1994, he joined the VLSI Systems and Design Methodologies (VSDM) division of the IMEC research center in Leuven, Belgium. Since 1996, he is heading IMEC's Multimedia Image Compression Systems group. This group focuses on the efficient design and implementation of embedded systems for advanced multimedia applications. Jan Bormans is the Belgian head of delegation for ISO/IEC's MPEG and SC29 standardization committees. He is also MPEG-21 requirements editor and chairman of the MPEG liaison group.

Fernando Pereira was born in Vermelha, Portugal in October 1962. He was graduated in Electrical and Computers Engineering by Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 1985. He received the M.Sc. and Ph.D. degrees in Electrical and Computers Engineering from IST, in 1988 and 1991, respectively.

He is currently Professor at the Electrical and Computers Engineering Department of IST. He is responsible for the participation of IST in many national and international research projects. He is a member of the Editorial Board and Area Editor on Image/Video Compression of the Signal Processing: Image Communication Journal and an Associate Editor of IEEE Transactions of Circuits and Systems for Video Technology, IEEE Transactions on Image Processing, and IEEE Transactions on Multimedia. He is a member of the Scientific and Program Committees of tens of international conferences and workshops. He has contributed more than 130 papers to journals and international conferences. He won the 1990 Portuguese IBM Award and an ISO Award for Outstanding Technical Contribution for his participation in the development of the MPEG-4 Visual standard, in October 1998.

He has been participating in the work of ISO/MPEG for many years, notably as the head of the Portuguese delegation, chairman of the MPEG Requirements group, and chairing many Ad Hoc Groups related to the MPEG-4 and MPEG-7 standards.

His current areas of interest are video analysis, processing, coding and description, and multimedia interactive services.



Peter List graduated in Applied Physics in 1985 and received the PH.D. in 1989 from the University of Frankfurt/Main, Germany.

Currently he is project manager at T-System Nova, the R&D Company of Deutsche Telekom. Since 1990 he has been with Deutsche Telekom, and has actively followed international standardization of video compression technologies in MPEG, ITU and several European Projects for about 14 years.

Detlev Marpe received the Diploma degree in mathematics with highest honors from the Technical University Berlin, Germany. He is currently a Project Manager in the Image Processing Department of the Fraunhofer-I nstitute for Telecommunications, Heinrich-Hertz-Institute (HHI), Berlin, Germany, where he is responsible for research projects in the area of video coding, image processing, and video streaming. Since 1997, he has been an active contributor to the ITU-T VCEG, ISO/IEC JPEG and ISO/IEC MPEG standardization activities for still image and video coding. During 2001–2003, he chaired the CABAC Ad-Hoc Group within the H.264/MPEG-4 AVC standardization effort of the ITU-T/ISO/IEC Joint Video Team. He has authored or co-authored more than 30 journal and conference papers in the fields of image and video coding, image processing and information theory, and he has written more than 40 technical contributions to various international standardization projects. He also holds several international patents. He is a member of IEEE and ITG (German Society of Information Technology). As a co-founder of *daViKo* GmbH, a Berlin-based start-up company involved in development of server-less multipoint videoconferencing products for Intranet or Internet collaboration, he received the Prime Prize of the 2001 Multimedia Start-up Competition founded by the German Federal Ministry of Economics and Technology.