The H.264/MPEG4 Advanced Video Coding Standard and its Applications

Detlev Marpe and Thomas Wiegand, Heinrich Hertz Institute (HHI),
Gary J. Sullivan, Microsoft Corporation

ABSTRACT

H.264/MPEG4-AVC is the latest video coding standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). H.264/MPEG4-AVC has recently become the most widely accepted video coding standard since the deployment of MPEG2 at the dawn of digital television, and it may soon overtake MPEG2 in common use. It covers all common video applications ranging from mobile services and videoconferencing to IPTV, HDTV, and HD video storage. This article discusses the technology behind the new H.264/MPEG4-AVC standard, focusing on the main distinct features of its core coding technology and its first set of extensions, known as the fidelity range extensions (FREXt). In addition, this article also discusses the current status of adoption and deployment of the new standard in various application areas.

INTRODUCTION AND HISTORICAL PERSPECTIVE

Digital video technology is enabling and generating ever new applications with a broadening range of requirements regarding basic video characteristics such as spatiotemporal resolution, chroma format, and sample accuracy. Application areas today range from videoconferencing over mobile TV and broadcasting of standard-/high-definition TV content up to very-high-quality applications such as professional digital video recording or digital cinema/large-screen digital imagery. Prior video coding standards such as MPEG2/H.262 [1], H.263 [2], and MPEG4 Part 2 [3] are already established in parts of those application domains. But with the proliferation of digital video into new application spaces such as mobile TV or high-definition TV broadcasting, the requirements for efficient representation of video have increased up to operation points where previously standardized video coding technology can hardly keep pace. Furthermore, more cost-efficient solutions in terms of bit rate vs. end-to-end reproduction quality are increasingly sought in traditional application areas of digital video as well.

Regarding these challenges, H.264/MPEG4 Advanced Video Coding (AVC) [4], as the latest entry of international video coding standards, has demonstrated significantly improved coding efficiency, substantially enhanced error robustness, and increased flexibility and scope of applicability relative to its predecessors [5]. A recently added amendment to H.264/MPEG4-AVC, the so-called fidelity range extensions (FREXt) [6], further broaden the application domain of the new standard toward areas like professional contribution, distribution, or studio/post production. Another set of extensions for scalable video coding (SVC) is currently being designed [7, 8], aiming at a functionality that allows the reconstruction of video signals with lower spatiotemporal resolution or lower quality from parts of the coded video representation (i.e., from partial bitstreams). The SVC project is planned to be finalized in January 2007. Also, multi-view video coding (MVC) capability has been successfully demonstrated using H.264/MPEG4-AVC [9], requiring almost no change to the technical content of the standard.

Rather than providing a comprehensive overview that covers all technical aspects of the H.264/MPEG4-AVC design, this article focuses on a few representative features of its core coding technology. After presenting some information about target application areas and the current status of deployment of the new standard into those areas, this article provides a high-level overview of the so-called video coding layer (VCL) of H.264/MPEG4-AVC. Being designed for efficiently representing video content, the VCL is complemented by the network abstraction layer (NAL), which formats the VCL representation and provides header information in a manner appropriate for conveyance by a variety of transport layers or storage media. A representative selection of innovative features of the video coding layer in H.264/MPEG4-AVC is described in more detail by putting emphasis on some selected FREXt-specific coding tools. Profile and level definitions of H.264/MPEG4-AVC are briefly discussed and finally a rate-distortion (R-D) performance comparison between H.264/MPEG4-AVC and MPEG2 video coding technology is presented.

0163-6804/06/$20.00 © 2006 IEEE

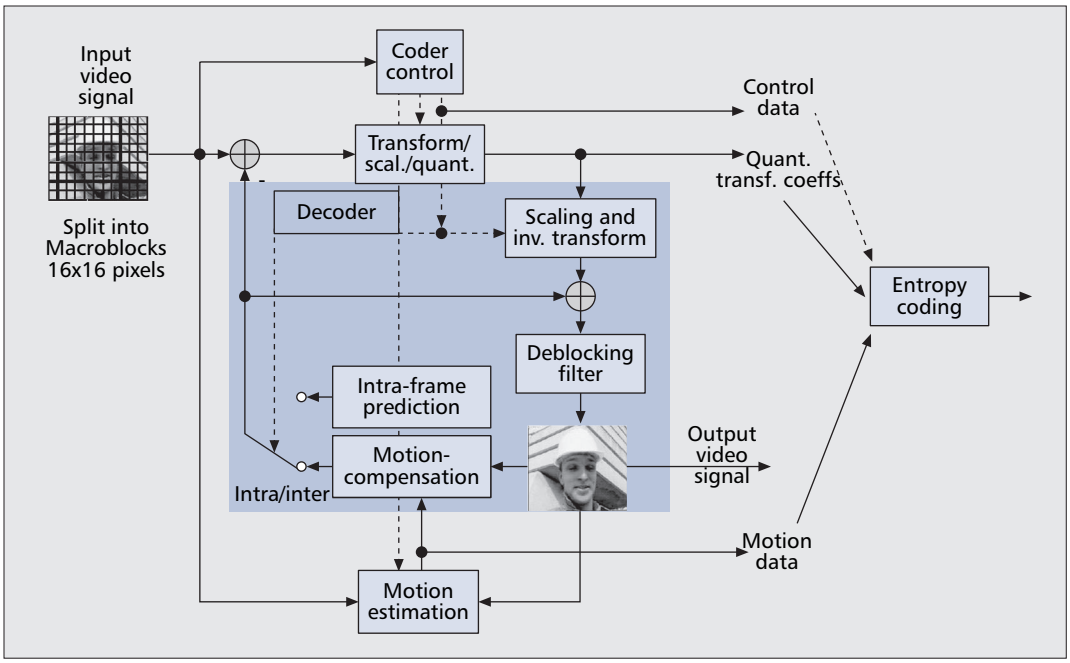


Figure 1. *Typical structure of an H.264/MPEG4-AVC video encoder.*

The video coding layer of H.264/MPEG4-AVC is similar in spirit to that of other video coding standards such as MPEG2 Video. In fact, it uses a fairly traditional approach consisting of a hybrid of block-based temporal and spatial prediction in conjunction with block-based transform coding.

Applications and Current Status of Deployments

As a generic, all-purpose video coding standard that is able to cover a broad spectrum of requirements from mobile phone to digital cinema applications within a single specification, H.264/MPEG4-AVC has received a great deal of recent attention from industry. Besides the classical application areas of videoconferencing and broadcasting of TV content (satellite, cable, and terrestrial), the improved compression capability of H.264/MPEG4-AVC enables new services and thus opens new markets and opportunities for the industry. As an illustration of this development, consider the case of "mobile TV" for the reception of audio-visual content on cell phones or portable devices, presently on the verge of commercial deployment. Several such systems for mobile broadcasting are currently under consideration, e.g.,

- Digital Multimedia Broadcasting (DMB) in South Korea
- Digital Video Broadcasting — Handheld (DVB-H), mainly in Europe and the United States
- Multimedia Broadcast/Multicast Service (MBMS), as specified in Release 6 of 3GPP

For such mobile TV services, improved video compression performance, in conjunction with appropriate mechanisms for error robustness, is key — a fact that is well reflected by the use of H.264/MPEG4-AVC (using the version 1 Baseline profile described below) together with forward error correction schemes in all of those mobile-broadcasting systems.

Another area that has attracted a lot of near-term industry implementation interest is the transmission and storage of HD content. Some indications of that trend are shown by the recent inclusion of H.264/MPEG4-AVC (using version 3, i.e., FREXt-related "High profile" described below) in important application standards or industry consortia specifications such as:

- The revised implementation guideline TS 101 154 of the Digital Video Broadcasting (DVB) organization
- The HD-DVD specification of the DVD Forum
- The BD specification of the Blu-Ray Disc Association (BDA)
- The International Telecommunication Union — Radiocommunication Standardization Sector (ITU-R) standards BT.1737 for HDTV contribution, distribution, satellite news gathering, and transmission, and BT.1687 for large-screen digital imagery for presentation in a theatrical environment

In addition, a number of providers of satellite television services (including DirecTV, BSkyB, Dish Network, Euro1080, Premiere, and ProSiebenSat.1) have recently announced or begun near-term deployments of H.264/MPEG4-AVC in so-called second generation HDTV delivery systems (often coupled with the new DVB-S2 satellite specification). At the time of writing this article, at least four single-chip solutions for HD decoding of H.264/MPEG4-AVC for set-top boxes are on the market by the semiconductor industry.

High-Level Overview of the Video Coding Layer

The video coding layer of H.264/MPEG4-AVC is similar in spirit to that of other video coding standards such as MPEG2 Video [1]. In fact, it uses a fairly traditional approach consisting of a hybrid of block-based temporal and spatial prediction in conjunction with block-based transform coding. Figure 1 shows an encoder block diagram for such a design.

A coded video sequence in H.264/MPEG4-

The typical encoding operation for a picture begins with splitting the picture into blocks of samples. The first picture of a sequence or a random access point is typically coded in Intra mode. For all remaining pictures of a sequence or between random access points, typically Inter coding is utilized.

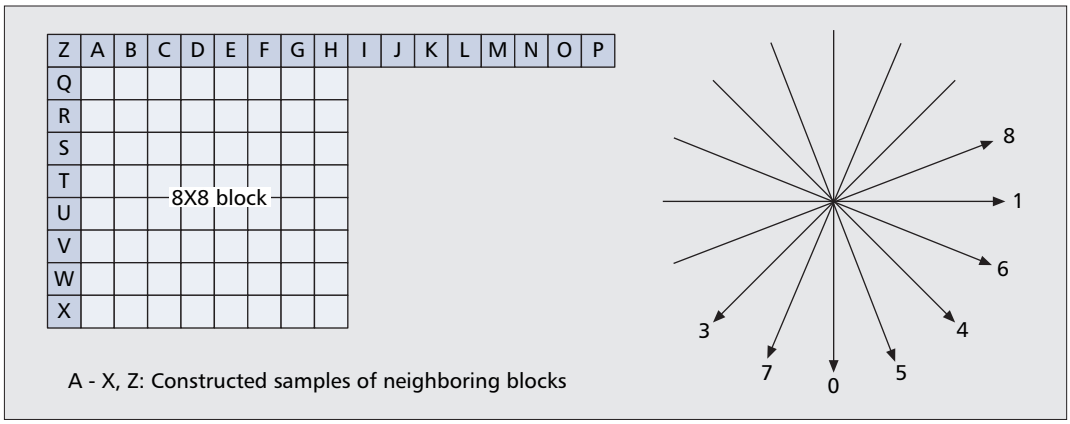


Figure 2. *Samples used for 8 × 8 spatial luma intra prediction (left), and directions of 4 × 4 and 8 × 8 spatial luma intra prediction modes 0, 1, and 3–8 (right).*

AVC consists of a sequence of coded pictures [4, 5]. A coded picture can represent either an entire *frame* or a single *field*, as was also the case for MPEG2 video. Generally, a frame of video can be considered to contain two interleaved fields: a top field and a bottom field. If the two fields of a frame were captured at different time instants, the frame is referred to as an *interlaced-scan* frame; otherwise, it is referred to as a *progressive-scan* frame.

The typical encoding operation for a picture begins with splitting the picture into blocks of samples. The first picture of a sequence or a random access point is typically coded in *Intra* (intra-picture) mode (i.e., without using any other pictures as prediction references). Each sample of a block in such an Intra picture is predicted using spatially neighboring samples of previously coded blocks. The encoding process chooses which neighboring samples are to be used for Intra prediction and how these samples are to be combined to form a good prediction, and sends an indication of its selection to the decoder.

For all remaining pictures of a sequence or between random access points, typically *Inter* (inter-picture) coding is utilized. Inter coding employs interpicture temporal prediction (motion compensation) using other previously decoded pictures. The encoding process for temporal prediction includes choosing motion data that identifies the reference pictures and spatial displacement vectors that are applied to predict the samples of each block.

The *residual* of the prediction (either Intra or Inter), which is the difference between the original input samples and the predicted samples for the block, is transformed. The transform coefficients are then scaled and approximated using scalar quantization. The quantized transform coefficients are entropy coded and transmitted together with the entropy-coded prediction information for either Intra- or Inter-frame prediction.

The encoder contains a model of the decoding process (shown as the shaded part of the block diagram in Fig. 1) so that it can compute the same prediction values computed in the decoder for the prediction of subsequent blocks in the current picture or subsequent coded pictures. The decoder inverts the entropy coding processes, performs the prediction process as indicated by the encoder using the prediction type information and motion data. It also inverse-scales and inverse-transforms the quantized transform coefficients to form the approximated residual and adds this to the prediction. The result of that addition is then fed into a deblocking filter, which provides the decoded video as its output.

Main Innovative Features of the Video Coding Layer

This section contains a more detailed description of the main building blocks of the H.264/MPEG4-AVC video coding layer sketched in the last section. The innovative nature of the characteristic features of those coding tools can be best described as having a substantially higher degree of diversification, sophistication, and adaptability than their counterparts in prior video coding standards. After presenting the rather traditional concept of how pictures are partitioned into smaller coding units below, some of the most representative innovations of the H.264/MPEG4-AVC video coding layer are introduced step by step, largely following the order of processing in the encoder as described in the previous section.

Subdivision of a Picture into Macroblocks and Slices

Each picture of a video sequence is partitioned into fixed size *macroblocks* that each cover a rectangular picture area of 16 × 16 samples of the luma component and, in the case of video in 4:2:0 chroma sampling format, 8 × 8 samples of each of the two chroma components. All luma and chroma samples of a macroblock are either spatially or temporally predicted, and the resulting prediction residual is represented using transform coding. Each color component of the prediction residual is subdivided into blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and entropy coded.

The macroblocks are organized in *slices*, which represent regions of a given picture that

can be decoded independent of each other. H.264/MPEG4-AVC supports five slice-coding types. The simplest is the *I* slice (where *I* stands for intra). In *I* slices all macroblocks are coded without referring to any other pictures of the video sequence. Prior coded images can be used to form a prediction signal for macroblocks of the predictive-coded *P* and *B* slice types (where *P* stands for predictive and *B* stands for bi-predictive). The remaining two slice types are *SP* (switching *P*) and *SI* (switching *I*) slices, which are specified for efficient switching between bitstreams coded at various bit rates [5].

A *picture* comprises the set of slices representing a complete frame or field. Splitting an interlaced-scan picture to create separate pictures for each field is especially efficient for random access purposes if the first field is coded using *I* slices and the second field is predicted from it using motion compensation. Furthermore, field-based coding is often utilized when the scene shows strong motion, as this leads to a reduced degree of statistical dependency between adjacent sample rows (because the alternate rows of an interlaced frame are captured at different time instants). In some scenarios parts of the frame are more efficiently coded in field mode, while others are more efficiently coded in frame mode. Hence, H.264/MPEG4-AVC also supports macroblock-adaptive switching between frame and field coding (MBAFF). For that, pairs of vertically contiguous macroblocks in a coded frame are categorized as either two frame-segmented (i.e., vertically spatially neighboring) or two field-segmented (i.e., vertically interleaved) macroblocks. The prediction processes and prediction residual coding are then conducted using the selected segmentation.

Spatial Intra Prediction

Each macroblock can be transmitted in one of several coding types depending on the slice coding type. In all slice coding types, at least two intra macroblock coding types are supported. All intra coding types in H.264/MPEG4-AVC rely on prediction of samples in a given block conducted in the spatial domain, rather than in the transform domain as has been the case in previous video coding standards. The types are distinguished by their underlying luma prediction block sizes of 4×4, 8×8 (FREXt only), and 16×16, whereas the intra prediction process for chroma samples operates in an analogous fashion but always with a prediction block size equal to the block size of the entire macroblock's chroma arrays. In each of those intra coding types, and for both luma and chroma, spatially neighboring samples of a given block that have already been transmitted and decoded are used as a reference for spatial prediction of the given block's samples. The number of encoder-selectable prediction modes in each intra coding type is either four (for chroma and 16×16 luma blocks) or nine (for 4×4 and 8×8 luma blocks).

As illustrated in Fig. 2 for the case of 8×8 spatial luma prediction — a type that is only supported by FREXt related profiles — luma values of each sample in a given 8×8 block are predicted from the values of neighboring decoded samples. In addition, as a distinguished feature of the 8×8 intra-coding type, the reference samples are smoothed by applying a low-pass filter prior to performing the actual prediction step. Eight different prediction directions plus an additional averaging (so-called DC) prediction mode (corresponding to mode 2 and not shown in Fig. 2) can be selected by the encoder. The 4×4 and 16×16 intra prediction types operate in a conceptually similar fashion except that they use different block sizes and do not include the smoothing filter.

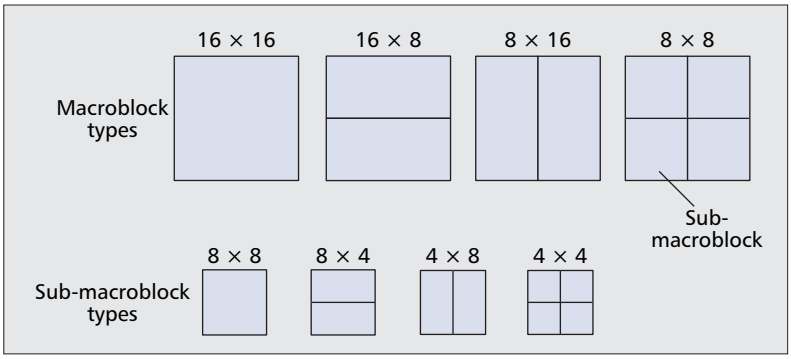


Figure 3. *Partitioning of a macroblock (top) and a sub-macroblock (bottom) for motion-compensated prediction.*

Motion-Compensated Prediction in *P* Slices

In addition to the intra macroblock coding types, various predictive or motion-compensated coding types are allowed in *P* slices. Each *P*-type macroblock is partitioned into fixed size blocks used for motion description. Partitionings with luma block sizes of 16×16, 16×8, 8×16, and 8×8 samples are supported by the syntax. When the macroblock is partitioned into four so-called sub-macroblocks each of size 8×8 luma samples, one additional syntax element is transmitted for each 8×8 sub-macroblock. This syntax element specifies whether the corresponding sub-macroblock is coded using motion-compensated prediction with luma block sizes of 8×8, 8×4, 4×8, or 4×4 samples. Figure 3 illustrates the partitioning.

The prediction signal for each predictive-coded $M \times N$ luma block is obtained by displacing a corresponding area of a previously decoded reference picture, where the displacement is specified by a translational motion vector and a picture reference index. Thus, if the macroblock is coded using four 8×8 sub-macroblocks, and each sub-macroblock is coded using four 4×4 luma blocks, a maximum of 16 motion vectors may be transmitted for a single *P*-slice macroblock. The motion vector precision is at the granularity of one quarter of the distance between luma samples. If the motion vector points to an integer-sample position, the prediction signal is formed by the corresponding samples of the reference picture; otherwise, the prediction signal is obtained using interpolation between integer-sample positions. The prediction values at half-sample positions are obtained by separable application of a one-dimensional six-tap finite impulse response (FIR) filter, and

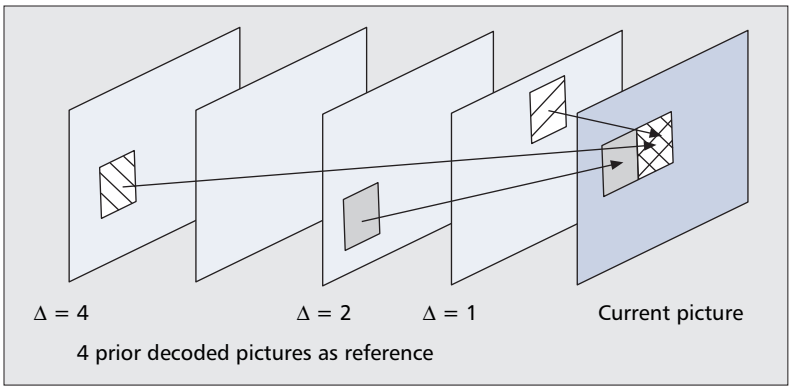


Figure 4. *Multiframe motion compensation. In addition to the motion vector, picture reference parameters (Δ) are also transmitted.*

prediction values at quarter-sample positions are generated by averaging samples at integer- and half-sample positions. The prediction values for the chroma components are obtained by bilinear interpolation.

H.264/MPEG4-AVC supports multi-picture motion-compensated prediction in a manner similar to what was known as enhanced reference picture selection in H.263 v. 3 [3]. That is, more than one prior coded picture can be used as reference for motion-compensated prediction. Figure 4 illustrates the concept which is also extended to *B* pictures as described below.

For multi-frame motion-compensated prediction, the encoder stores decoded reference pictures in a multi-picture buffer. The decoder replicates the multi-picture buffer of the encoder according to the reference picture buffering type and memory management control operations (MMCO) specified in the bitstream. Unless the size of the multi-picture buffer is set to one picture, the index at which the reference picture is located inside the multi-picture buffer has to be signaled. The reference index parameter is transmitted for each motion-compensated 16×16, 16×8, or 8×16 macroblock partition or 8×8 sub-macroblock.

In addition to the macroblock modes described above, a *P*-slice macroblock can also be coded in the so-called skip mode. For this mode, neither a quantized prediction error signal nor a motion vector or reference index parameter are transmitted. The reconstructed signal is computed in a manner similar to the prediction of a macroblock with partition size 16×16 and fixed reference picture index equal to 0. In contrast to previous video coding standards, the motion vector used for reconstructing a skipped macroblock is inferred from motion properties of neighboring macroblocks rather than being inferred as zero (i.e., no motion).

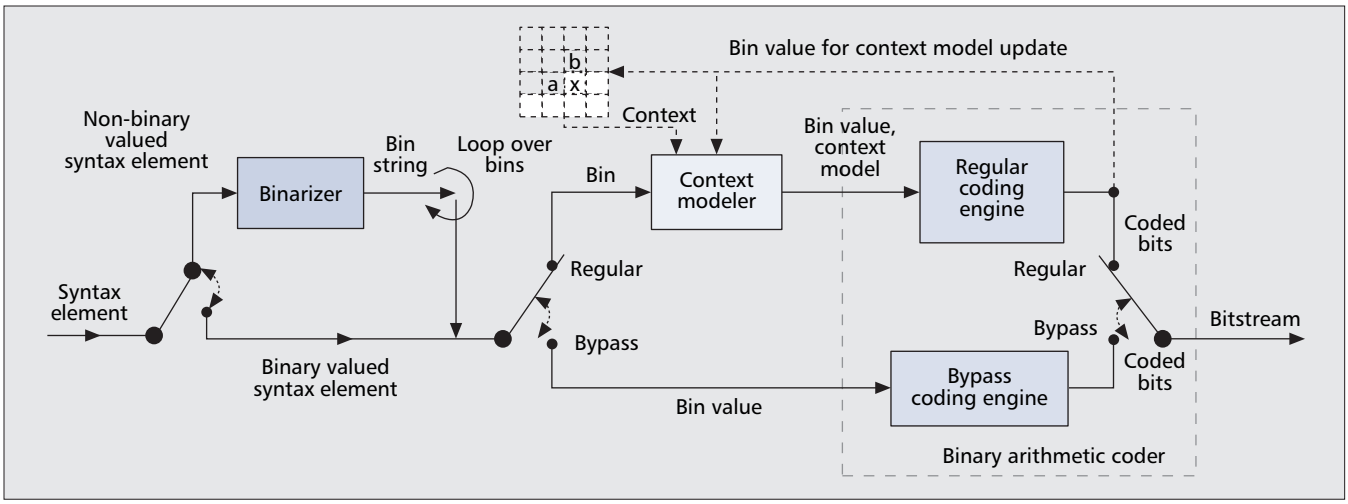
Motion-Compensated Prediction in *B* Slices

In comparison to prior video coding standards, the concept of *B* slices in H.264/MPEG4-AVC is generalized in several ways [5]. For example, unlike in MPEG2 video, B pictures themselves can be used as reference pictures for motion-compensated prediction. Thus, the only substantial difference between *B* and *P* slices in H.264/MPEG4-AVC is that *B* slices are coded in a manner in which some macroblocks or blocks may use a weighted average of two distinct motion-compensated prediction values for building the prediction signal. However, as another extension of the corresponding functionality beyond MPEG2 video, this does not imply the restriction to the case of using a superposition of forward and backward prediction signals in the classical sense. In fact, the concept of *generalized B pictures* in H.264/MPEG4-AVC allows any arbitrary pair of reference pictures to be utilized for the prediction of each region (as exemplified in Fig. 4). For that purpose, two distinct ways of indexing the multi-picture buffer are maintained for *B* slices, which are referred to as the first ("*list 0*") and second ("*list 1*") reference picture lists, respectively. The ordering of these lists is signaled by the encoder. It is also worth noting that by decoupling the ordering of pictures for display and referencing purposes in H.264/MPEG4-AVC, greater flexibility can be achieved, particularly with respect to the control of the structural decoding delay caused by using reference pictures that are displayed later than other pictures that use them as references in the decoding process. This flexibility has been shown to have increasing importance over time, including its use as a fundamental part of the new SVC and upcoming MVC extensions [7–9].

Depending on which reference picture list is used for forming the prediction signal, three different types of interpicture prediction are distinguished in *B* slices: *list 0*, *list 1*, and *bi-predictive*, where the latter uses a superposition of list 0 and list 1 prediction signals and is the key feature provided by *B* slices. With a similar partitioning as specified for *P* slices, the three different interpicture prediction types in *B* slices can be chosen separately for each macroblock partition or sub-macroblock partition. Additionally, *B*-slice macroblocks or sub-macroblocks can also be coded in so-called direct mode without the need to transmit any additional motion information. If no prediction residual data are transmitted for a direct-coded macroblock, it is also referred to as *skipped*, and skipped macroblocks can be indicated very efficiently as for the skip mode in *P* slices [10].

Transform, Scaling, and Quantization

As already noted above, H.264/MPEG4-AVC also uses transform coding of the prediction residual. However, in contrast to prior video coding standards. such as MPEG2 or H.263. which use a 2D discrete cosine transform (DCT) of size 8×8, H.264/MPEG4-AVC specifies a set of integer transforms of different block sizes. In all version 1 related profiles, a 4×4 integer transform [5] is applied to both the luma and chroma components of the prediction residual signal. An additional $M \times N$ transform stage is further applied to all resulting DC coefficients in the case of the luma component of a macroblock that is coded using the 16×16 intra-coding type (with $N=M=4$) as well as in the case of both chroma components (with the values of $N, M \in \{2,4\}$ depending on the chroma format). For these additional transform stages, separable



■ **Figure 5.** *CABAC block diagram.*

combinations of the four-tap Hadamard transform and two-tap Haar/Hadamard transform are applied.

Besides the important property of low computational complexity, the use of those small block-size transforms in H.264/MPEG4-AVC has the advantage of significantly reducing ringing artifacts. For high-fidelity video, however, the preservation of smoothness and texture generally benefits from a representation with longer basis functions. A better trade-off between these conflicting objectives can be achieved by making use of the 8×8 integer transform specified in the FRExt amendment as an additional transform type for coding the luma residual signal. This 8×8 block transform is a close approximation of the 2D 8×8 DCT, and provides the benefit of allowing efficient implementations in integer arithmetic [11, 12]. In fact, all integer transforms in H.264/MPEG4-AVC as well as their corresponding inverse transforms can be implemented in a very cost-efficient way since only shift and add operations in $(8 + b)$-bit arithmetic precision are required for processing b-bit input video.

As an additional degree of freedom in the FRExt profiles, the encoder has the choice between using the 4×4 or 8×8 transform in order to adapt the representation of the luma residual signal to its specific characteristics on a macroblock-by-macroblock basis. This adaptive choice is coupled together with related parts of the decoding process; for example, by disallowing use of the 8×8 transform when the prediction block size is smaller than 8×8.

For the quantization of transform coefficients, H.264/MPEG4-AVC uses uniform-reconstruction quantizers (URQs). One of 52 quantizer step size scaling factors is selected for each macroblock by a quantization parameter (QP). The scaling operations are arranged so that there is a doubling in quantization step size for each increment of six in the value of QP. The quantized transform coefficients of a block generally are scanned in a zig-zag fashion and further processed using the entropy coding methods described below. In addition to the basic step-size control, the FRExt amendment also supports encoder-specified scaling matrices for a perceptual tuned, frequency-dependent quantization (a capability similar to that found in MPEG2 video).

In-Loop Deblocking Filter

Due to coarse quantization at low bit rates, block-based coding typically results in visually noticeable discontinuities along the block boundaries. If no further provision is made to deal with this, these artificial discontinuities may also diffuse into the interior of blocks by means of the motion-compensated prediction process. The removal of such blocking artifacts can provide a substantial improvement in perceptual quality. For that purpose, H.264/MPEG4-AVC defines a deblocking filter that operates within the predictive coding loop, and thus constitutes a required component of the decoding process. The filtering process exhibits a high degree of content adaptivity on different levels, from the slice level along the edge level down to the level of individual samples. As a result, the blockiness is reduced without much affecting the sharpness of the content. Consequently, the subjective quality is significantly improved. At the same time, the filter reduces bit rate by typically 5–10 percent while producing the same objective quality as the non-filtered video.

Entropy Coding

In H.264/MPEG4-AVC, many syntax elements are coded using the same highly-structured infinite-extent variable-length code (VLC), called a zero-order exponential-Golomb code. A few syntax elements are also coded using simple fixed-length code representations. For the remaining syntax elements, two types of entropy coding are supported.

When using the first entropy-coding configuration, which is intended for lower-complexity (esp. software-based) implementations, the exponential-Golomb code is used for nearly all syntax elements except those of quantized transform coefficients, for which a more sophisticated method called context-adaptive variable length coding (CAVLC) is employed. When using CAVLC, the encoder switches between different

The Baseline profile was targeted at applications in which a minimum of computational complexity and a maximum of error robustness are required, whereas the Main profile was aimed at applications that require a maximum of coding efficiency, with somewhat less emphasis on error robustness.

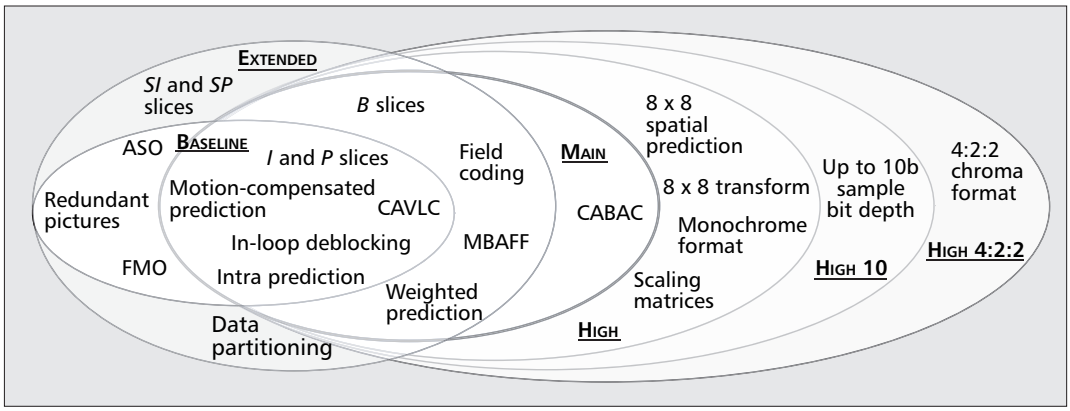


Figure 6. *Illustration of H.264/MPEG4-AVC profiles.*

VLC tables for various syntax elements, depending on the values of the previously transmitted syntax elements in the same slice. Since the VLC tables are designed to match the conditional probabilities of the context, the entropy coding performance is improved from that of schemes that do not use context-based adaptivity.

The entropy coding performance is further improved if the second configuration is used, which is referred to as context-based adaptive binary arithmetic coding (CABAC) [5]. As depicted in Fig. 5, the CABAC design is based on three components: binarization, context modeling, and binary arithmetic coding. Binarization enables efficient binary arithmetic coding by mapping nonbinary syntax elements to sequences of bits referred to as *bin strings*. The bins of a bin string can each be processed in either an arithmetic coding mode or a *bypass* mode. The latter is a simplified coding mode that is chosen for selected bins such as sign information or lesser-significance bins in order to speed up the overall decoding (and encoding) processes. The arithmetic coding mode provides the largest compression benefit, where a bin may be context-modeled and subsequently arithmetic encoded. As a design decision, in most cases only the most probable bin of a syntax element is supplied with external context modeling, which is based on previously decoded (encoded) bins. The compression performance of the arithmetic-coded bins is optimized by adaptive estimation of the corresponding (context-conditional) probability distributions. The probability estimation and the actual binary arithmetic coding are conducted using a multiplication-free method that enables efficient implementations in hardware and software. Compared to CAVLC, CABAC can typically provide reductions in bit rate of 10–20 percent for the same objective video quality when coding SDTV/HDTV signals.

Profiles and Levels

Profiles and levels specify conformance points that provide interoperability between encoder and decoder implementations within applications of the standard and between various applications that have similar functional requirements. A profile defines a set of syntax features for use in generating conforming bitstreams, whereas a level places constraints on certain key parameters of the bitstream such as maximum bit rate and maximum picture size. All decoders conforming to a specific profile and level must support all features included in that profile when constrained as specified for the level. Encoders are not required to make effective use of any particular set of features supported in a profile and level but must not violate the syntax feature set and associated constraints. This implies in particular that conformance to any specific profile and level, although it ensures interoperability with decoders, does not provide any guarantees of end-to-end reproduction quality. Figure 6 illustrates the current six profiles of H.264/MPEG4-AVC and their corresponding main features, as further discussed below.

The Baseline, Main, and Extended Profile (Version 1)

In the first version of H.264/MPEG4-AVC three profiles were defined: the Baseline, Extended, and Main profiles. The Baseline profile supports all features in H.264/MPEG4-AVC, v. 1 (2003), except the following three feature sets:

- Set 1: *B* slices, field picture coding, macroblock-adaptive switching between frame and field coding (MBAFF), and weighted prediction
- Set 2: CABAC
- Set 3: *SP* and *SI* slices, and slice data partitioning

The first and second of these three feature sets is supported by the Main profile (MP), in addition to the features supported by the Baseline profile except for the FMO feature and some other enhanced error resilience features [4]. The Extended profile supports all features of the Baseline profile, and the first and third above sets of features, but not CABAC.

Roughly speaking, the Baseline profile was targeted at applications in which a minimum of computational complexity and a maximum of error robustness are required, whereas the Main profile was aimed at applications that require a maximum of coding efficiency, with somewhat less emphasis on error robustness. The Extended profile was designed to provide a compromise between the Baseline and Main profile capabilities with an additional focus on the specific needs

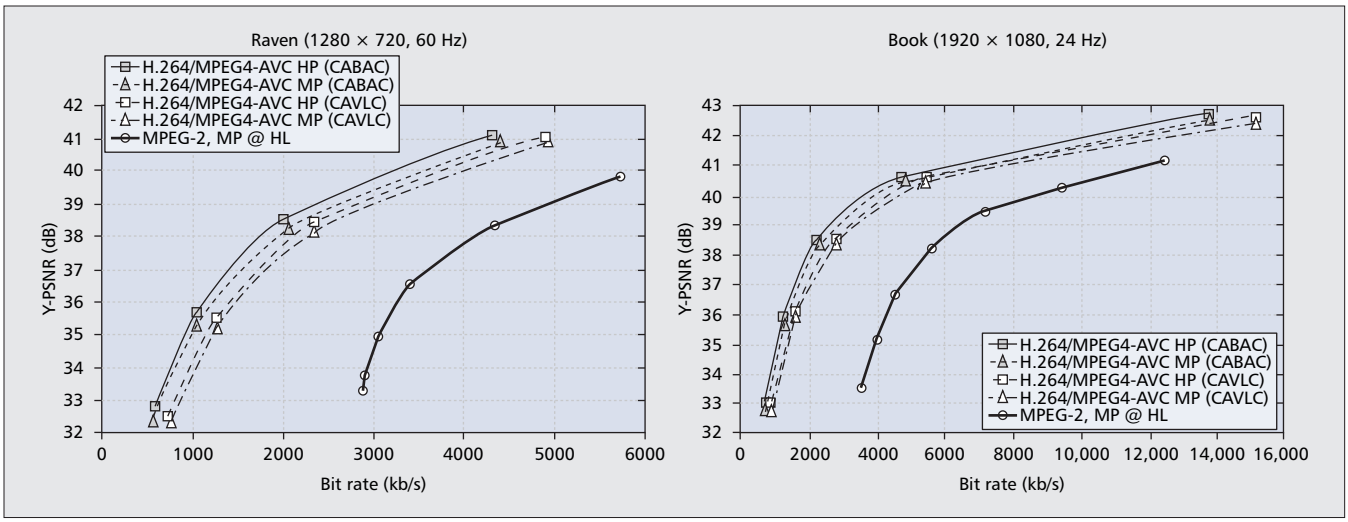


Figure 7. *Left: Objective performance for the "Raven" sequence (left) and "Book" sequence (right) comparing H.264/MPEG4-AVC HP, MP (both using CABAC and CAVLC), and MPEG2 MP@HL.*

of video streaming applications, and further added robustness to errors and packet losses.

High Profiles in the FRExt Amendment (Version 3)

As depicted in Fig. 6, the H.264/MPEG4-AVC FRExt amendment specifies, in addition to the three profiles of v. 1, a set of three additional profiles constructed as nested sets of capabilities built on top of the Main profile. As their common intersection, the High profile (HP) contains the most relevant FRExt tools for further improving coding efficiency. Relative to the Main profile, these tools imply only a moderate (if any) increase in complexity in terms of both implementation and computational costs (at the decoder side). Therefore, the High profile, with its restriction to 8-bit video in 4:2:0 chroma format, has overtaken the Main profile for prospective applications of H.264/MPEG4-AVC in typical SD and HD consumer applications. Two other profiles, called the *High 10* and *High 4:2:2* profiles, further extend the capability of the standard to include more demanding applications requiring higher sample precision (up to 10 b/sample) and higher chroma formats (up to 4:2:2).[1]

R-D Performance

A comparison of the compression performance achievable with H.264/MPEG4-AVC v. 1 profiles can be found in [5]. Here we also focus on a demonstration of the additional benefit that can be obtained by using some of the HP-specific coding tools. More specifically, we have evaluated the gain in objective performance for the HP-specific 8 × 8 coding tools in terms of objective performance. For that purpose, we have performed a series of coding simulations by using a test set of seven progressive HD sequences with different characteristics and different spatiotemporal resolutions (four 720p sequences with 1280 × 720 @ 60 Hz and three 1080p sequences with 1920 × 1080 @ 24 Hz). The coding simulations were carried out using the H.264/MPEG4-AVC reference software encoder (version JM 9.2) and an MPEG2 Main profile (MP@HL) conforming encoder. To provide a fair comparison, both encoders have been controlled using the same R-D optimized encoding strategy [5]. For both encoders, an *I*-frame refresh was performed every 500 ms, and two non-reference *B* pictures have been inserted between each two successive *P* pictures. Full-search motion estimation was performed with a search range of ±32 integer pixels. For H.264/MPEG4-AVC up to three reference frames were used.

As an example of the outcome of our experiments, Fig. 7 shows the R-D curves for the 720p "Raven" (left) and 1080p "Book" (right) sequences comparing H.264/MPEG4-AVC HP (including 8 × 8 coding tools), MP with both CABAC and CAVLC, and MPEG2. Since these sequences are characterized by predominantly highly textured content, relatively large gains can be obtained in favor of HP due to the better frequency selectivity of the 8 × 8 luma transform. Averaged over the whole HD test set and a quality range of 33–39 dB peak signal-to-noise ratio (PSNR), HP achieves bit rate savings of about 10 percent relative to MP (both using CABAC), as shown in Table 1. If, however, the 8 × 8 tool set of HP is used in conjunction with CAVLC, an average loss of about 18 percent is observed relative to HP using CABAC, which means that the CAVLC-driven HP leads, on average, to objectively lower performance than that measured for the CABAC-driven MP (Table 1).

In comparison with MPEG2, the H.264/MPEG4-AVC High profile coder (with 8 × 8 coding tools and CABAC enabled) achieves average bit rate savings of about 59 percent when measured over the entire test set and investigated PSNR range.

Conclusions

The new H.264/MPEG4-AVC video coding standard was developed and standardized collaboratively by both the ITU — Telecommunication

[1] *Originally, the FRExt amendment included another, so-called high 4:4:4 profile which, at the time of writing this article, is in the process of being removed from the specification, as the JVT plans to replace it with at least two new profiles of a somewhat different design that are yet to be finalized.*

	Average bit rate savings relative to:		
Coder	H.264/MPEG4-AVC HP using CAVLC	H.264/MPEG4-AVC MP using CABAC	MPEG2 MP@HL
H.264/MPEG4-AVC HP using CABAC	17.9%	9.9%	58.8%

Table 1. *Average bit rate savings for H.264/MPEG4-AVC HP using CABAC entropy coding relative to H.264/MPEG4-AVC HP using CAVLC, H.264/MPEG4-AVC MP using CABAC, and MPEG2 MP@HL.*

Standardization Sector (ITU-T) VCEG and ISO/IEC MPEG organizations. H.264/MPEG4-AVC represents a number of advances in standardized video coding technology, in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application domains. Its video coding layer design is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important innovations relative to prior standards. We summarize some of the important differences thusly:

- Enhanced motion-compensated prediction and spatial intra prediction capabilities
- Use of 4×4 and 8×8 (FREXt only) transforms in integer precision
- Content-adaptive in-loop deblocking filter
- Enhanced entropy coding methods

When used well together, the features of the new design provide significant bit rate savings for equivalent perceptual quality relative to the performance of prior standards. This is especially true for use of the High profile related coding tools.

Acknowledgment

The authors thank the experts of ITU-T VCEG, ISO/IEC MPEG, and the ITU-T/ISO/IEC Joint Video Team for their contributions. The authors would also like to thank the anonymous reviewers for their helpful comments and suggestions.

Further Reading

Further information and documents of the JVT project are available online at http://ftp3.itu.ch/av-arch/jvt-site/. The reader interested in individual technical subjects within the scope of version 1 of H.264/MPEG4-AVC is referred to a special journal issue on H.264/MPEG4-AVC [5]. Additional information about FREXt-specific technical aspects can be found in [13, 14], while [15] includes some background information about the history and development of the new standard as well as information related to the recent deployment and adoption status.

References

[1] ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG2), "Generic Coding of Moving Pictures and Associated Audio Information — Part 2: Video," Nov. 1994.
[2] ITU-T Rec. H.263, "Video Coding for Low Bit Rate Communication," v1, Nov. 1995; v2, Jan. 1998; v3, Nov. 2000.
[3] ISO/IEC JTC 1, "Coding of Audio-Visual Objects — Part 2: Visual," ISO/IEC 14496-2 (MPEG4 Visual Version 1), April 1999; Amendment 1 (Version 2), Feb. 2000; Amendment 4 (streaming profile), Jan. 2001.
[4] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), "Advanced Video Coding for Generic Audiovisual Services," v1, May, 2003; v2, Jan. 2004; v3 (with FREXt), Sept. 2004; v4, July 2005.
[5] A. Luthra, G. J. Sullivan, and T. Wiegand, Eds., Special issue on the "H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 13, no. 7, July 2003.
[6] G. J. Sullivan *et al.*, "Draft text of H.264/AVC Fidelity Range Extensions Amendment," ISO/IEC MPEG and ITU-T VCEG, JVT-L047, Redmond, WA, July 2004.
[7] T. Wiegand *et al.*, "Joint Draft 5: Scalable Video Coding," ISO/IEC MPEG and ITU-T VCEG, Doc. JVT-R201, Bangkok, Thailand, Jan. 2006.
[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable H.264/MPEG4-AVC Extension," to be presented, *IEEE Int'l. Conf. Image Processing*, Atlanta, GA, Oct. 2006.
[9] K. Müller *et al.*, "Multi-View Video Coding Based on H.264/AVC Using Hierarchical B-Frames," *Proc. PCS 2006,* Beijing, China, Apr. 2006.
[10] A. M. Tourapis *et al.*, "Direct Mode Coding for Bipredicitve Slices in the H.264 Standard," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 15, no. 1, Jan. 2005, pp. 119–26.
[11] S. Gordon, D. Marpe, and T. Wiegand, "Simplified Use of 8x8 Transforms," ISO/IEC MPEG and ITU-T VCEG, JVT-K028, Munich, Germany, Mar. 2004.
[12] F. Bossen, "ABT Cleanup and Complexity Reduction," ISO/IEC MPEG and ITU-T VCEG, JVT-E087, Geneva, Switzerland, Oct. 2002.
[13] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas," *IEEE Int'l. Conf. Image Processing*, vol. 1, Sept. 2005, pp. 593–96.
[14] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *SPIE Annual Conf. Apps. of Digital Image Processing XXVII*, Special Session on Advances in the New Emerging Standard H.264/AVC, Aug., 2004, pp. 454–74.
[15] G. J. Sullivan, "The H.264/MPEG4-AVC Video Coding Standard and Its Deployment Status," *SPIE Conf. Visual Commun. and Image Processing*, Beijing, China, July 2005.

Biographies

Detlev Marpe [M'00] received a Dr.-Ing. degree from the University of Rostock, Germany, in 2005, and a Dipl.-Math. degree (with highest honors) from the Technical University of Berlin (TUB), Germany, in 1990. From 1991 to 1993 he was a research and teaching assistant in the Department of Mathematics at TUB. Since 1994 he has been involved in several industrial and research projects in the area of still image coding, image processing, video coding, and video streaming. In 1999 he joined the Fraunhofer Institute for Telecommunications — Heinrich-Hertz-Institute (HHI), Berlin, Germany, where as a project manager in the Image Processing Department he is currently responsible for research projects focused on the development of advanced video coding and video transmission technologies. He has published more than 40 journal and conference articles in the area of image and video processing, and holds several international patents in this field. He has been involved in ITU-T and ISO/IEC standardization activities for still image and video coding, to which he has contributed about 50 input documents. From 2001 to 2003, as an Ad Hoc Group Chairman in the Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, he was responsible for the development of the CABAC entropy coding scheme within the H.264/MPEG4-AVC standardization project. He also served as co-editor of the H.264/MPEG4-AVC FREXt Amendment in 2004. As a co-founder of daViKo GmbH, a Berlin-based startup company involved in the development of serverless multipoint videoconferencing products for intranet or Internet collaboration, he received the Prime Prize of the 2001 Multimedia Startup Competition founded by the German Federal Ministry of Economics and Technology. In 2004 he received the Fraunhofer Prize for outstanding scientific achievements in solving application related problems and the ITG Award of the German Society for Information Technology. His current research interests include still image and video coding, image and video communication, as well as computer vision and information theory.

Thomas Wiegand [M'05] is head of the Image Communication Group in the Image Processing Department of the Fraunhofer Institute for Telecommunications — HHI. He received a Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Germany, in 1995 and a Dr.-Ing. degree from the University of Erlan-

gen-Nuremberg, Germany, in 2000. From 1993 to 1994 he was a visiting researcher at Kobe University, Japan. In 1995 he was a visiting scholar at the University of California at Santa Barbara, where he started his research on video compression and transmission. Since then he has published several conference and journal papers on the subject and has contributed successfully to the ITU-T Video Coding Experts Group (ITU-T SG16 Q.6 — VCEG)/ISO/IEC JTC1/SC29/WG11 — MPEG) Joint Video Team (JVT) standardization efforts and holds various international patents in this field. From 1997 to 1998 he was a visiting researcher at Stanford University, California. In October 2000 he was appointed Associate Rapporteur of ITU-T VCEG. In December 2001 he was appointed Associated Rapporteur/Co-Chair of the JVT. In February 2002 he was appointed editor of the H.264/AVC video coding standard. In January 2005 he was appointed Associate Chair of MPEG Video. In 1998 he received the SPIE VCIP Best Student Paper Award. In 2004 he received the Fraunhofer Prize for outstanding scientific achievements in solving application related problems and the ITG Award of the German Society for Information Technology. Since January 2006 he is an Associate Editor of *IEEE Transactions on Circuits and Systems for Video Technology*. His research interests include image and video compression, communication and signal processing, as well as vision and computer graphics.

GARY J. SULLIVAN (S'83-M'91-SM'01-F'06) received B.S. and M.Eng. degrees in electrical engineering from the University of Louisville J.B. Speed School of Engineering,Kentucky, in 1982 and 1983, respectively. He received Ph.D. and Engineer degrees in electrical engineering from the University of California, Los Angeles, in 1991. He is the ITU-T rapporteur/chairman of the ITU-T Video Coding Experts Group (VCEG), a co-chairman of the ISO/IEC Moving Picture Experts Group (MPEG), and a co-chairman of the Joint Video Team (JVT), which is a joint project between the VCEG and MPEG organizations. He has led ITU-T VCEG (ITU-T Q.6/SG16) since 1996 and is also the ITU-T video liaison representative to MPEG. In MPEG (ISO/IEC JTC1/SC29/WG11), in addition to his current service as a co-chair of its video work, he also served as the chairman of MPEG video from March 2001 to May 2002. In the JVT he was the JVT chairman for the development of the next-generation H.264/MPEG4-AVC video coding standard and its fidelity-range extensions (FREXt), and is now its co-chairman for the development of the scalable video coding (SVC) extensions. He received the Technical Achievement award of the International Committee on Technology Standards (INCITS) in 2005 for his work on H.264/MPEG4-AVC and other video standardization topics. He holds the position of video architect in the Core Media Processing Team in the Windows Digital Media division of Microsoft Corporation. At Microsoft he also designed and remains lead engineer for the DirectX® Video Acceleration API/DDI video decoding feature of the Microsoft Windows® operating system platform. Prior to joining Microsoft in 1999, he was the manager of communications core research at PictureTel Corporation, the quondam world leader in videoconferencing communication. He was previously a Howard Hughes Fellow and member of technical staff in the Advanced Systems Division of Hughes Aircraft Corporation, and a terrain-following radar system software engineer for Texas Instruments. His research interests and areas of publication include image and video compression, rate-distortion optimization, motion estimation and compensation, scalar and vector quantization, and error-/packet-loss-resilient video coding.