The emerging H.264/AVC standard

Ralf Schäfer, Thomas Wiegand and Heiko Schwarz
Heinrich Hertz Institute, Berlin, Germany

H.264/AVC is the current video standardization project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The main goals of this standardization effort are to develop a simple and straightforward video coding design, with enhanced compression performance, and to provide a "network-friendly" video representation which addresses "conversational" (video telephony) and "non-conversational" (storage, broadcast or streaming) applications.

H.264/AVC has achieved a significant improvement in the rate-distortion efficiency – providing, typically, a factor of two in bit-rate savings when compared with existing standards such as MPEG-2 Video.

The MPEG-2 video coding standard [1], which was developed about 10 years ago, was the enabling technology for all digital television systems worldwide. It allows an efficient transmission of TV signals over satellite (DVB-S), cable (DVB-C) and terrestrial (DVB-T) platforms. However, other transmission media such as xDSL or UMTS offer much smaller data rates. Even for DVB-T, there is insufficient spectrum available – hence the number of programmes is quite limited, indicating a need for further improved video compression.

In 1998, the *Video Coding Experts Group* (VCEG – ITU-T SG16 Q.6) started a project called H.26L with the target to double the coding efficiency when compared with any other existing video coding standard. In December 2001, VCEG and the *Moving Pictures Expert Group* (MPEG – ISO/IEC JTC 1/SC 29/WG 11) formed the *Joint Video Team* (JVT) with the charter to finalize the new video coding standard H.264/AVC [2].

The H.264/AVC design covers a **Video Coding Layer** (VCL), which efficiently represents the video content, and a **Network Abstraction Layer** (NAL), which formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by particular transport layers or storage media.

The VCL design – as in any prior ITU-T and ISO/IEC JTC1 standard since H.261 [2] – follows the so-called *block-based hybrid* video-coding approach. The basic source-coding algorithm is a hybrid of *inter-picture prediction*, to exploit the temporal statistical dependencies, and *transform coding of the prediction residual* to exploit the spatial statistical dependencies. There is no single coding element in the VCL that provides the majority of the dramatic improvement in compression efficiency, in relation to prior video coding standards. Rather, it is the plurality of smaller improvements that add up to the significant gain.

The next section provides an overview of the H.264/AVC design. The Profiles and Levels specified in the current version of H.264/AVC [2] are then briefly described, followed by a comparison of H.264/AVC Main profile with the profiles of prior coding standards, in terms of rate-distortion efficiency. Based on the study of rate-distortion performance, various new business opportunities are delineated, followed by a report on existing implementations.

Technical overview of H.264/AVC

The H.264/AVC design [2] supports the coding of video (in 4:2:0 chroma format) that contains either progressive or interlaced frames, which may be mixed together in the same sequence. Generally, a frame of video contains two interleaved fields, the top and the bottom field. The two fields of an interlaced frame, which are separated in time by a field period (half the time of a frame period), may be coded separately as two field pictures or together as a frame picture. A progressive frame should always be coded as a single frame picture; however, it is still considered to consist of two fields at the same instant in time.

Network abstraction layer

The VCL, which is described in the following section, is specified to represent, efficiently, the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance by the transport layers or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and bitstream delivery is identical – except that each NAL unit can be preceded by a start code prefix in a bitstream-oriented transport layer.

Video coding layer

The video coding layer of H.264/AVC is similar in spirit to other standards such as MPEG-2 Video. It consists of a hybrid of temporal and spatial prediction, in conjunction with transform coding. *Fig. 1* shows a block diagram of the video coding layer for a macroblock.

In summary, the picture is split into blocks. The first picture of a sequence or a random access point is typically "Intra" coded, i.e., without using information other than that contained in the picture itself. Each sample

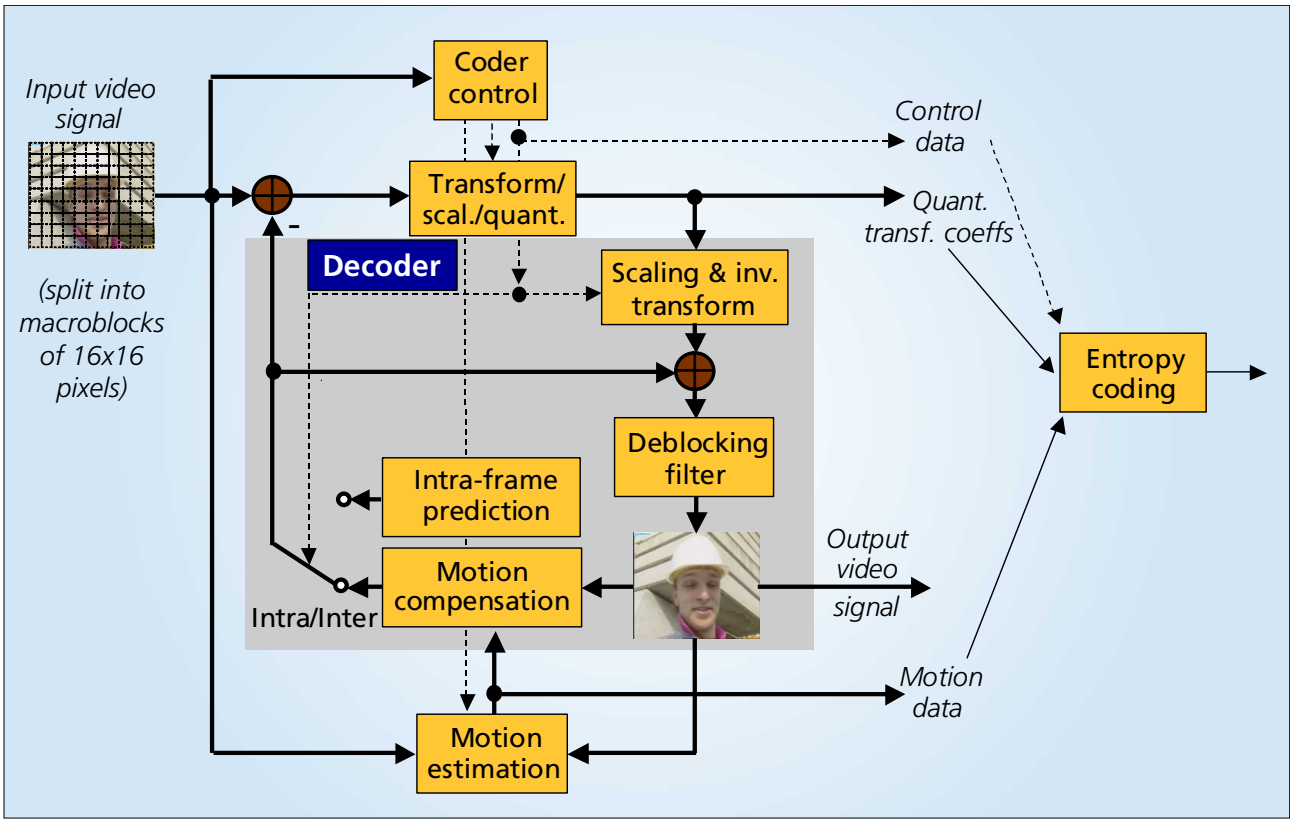

Figure 1
Basic coding structure of H.264/AVC for a macroblock

of a block in an Intra frame is predicted using spatially neighbouring samples of previously coded blocks. The encoding process chooses which and how neighbouring samples are used for Intra prediction, which is simultaneously conducted at the encoder and decoder using the transmitted Intra prediction side information.

For all remaining pictures of a sequence or between random access points, typically "Inter" coding is used. Inter coding employs prediction (motion compensation) from other previously decoded pictures. The encoding process for Inter prediction (motion estimation) consists of choosing motion data, comprising the reference picture, and a spatial displacement that is applied to all samples of the block. The motion data which are transmitted as side information are used by the encoder and decoder to simultaneously provide the Inter prediction signal.

The residual of the prediction (either Intra or Inter) – which is the difference between the original and the predicted block – is transformed. The transform coefficients are scaled and quantized. The quantized transform coefficients are entropy coded and transmitted together with the side information for either Intra-frame or Inter-frame prediction.

The encoder contains the decoder to conduct prediction for the next blocks or the next picture. Therefore, the quantized transform coefficients are inverse scaled and inverse transformed in the same way as at the decoder side, resulting in the decoded prediction residual. The decoded prediction residual is added to the prediction. The result of that addition is fed into a deblocking filter which provides the decoded video as its output.

A more detailed description of the technical contents of H.264 is given below. Readers less interested in technical details may want to skip these sections and continue by reading the section on "Profiles and levels" *(see page 8)*.

Subdivision of a picture into macroblocks

Each picture of a video, which can either be a frame or a field, is partitioned into fixed-size macroblocks that cover a rectangular picture area of 16×16 samples of the luma component and 8×8 samples of each of the two

Abbreviations

3G	3rd Generation mobile communications
3GPP	3rd Generation Partnership Project
16-QAM	16-state Quadrature Amplitude Modulation
ASP	(MPEG-4) Advanced Simple Profile
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CIF	Common Intermediate Format
DCT	Discrete Cosine Transform
DVB	Digital Video Broadcasting
DVB-C	DVB - Cable
DVB-S	DVB - Satellite
DVB-T	DVB - Terrestrial
FIR	Finite Impulse Response
FMO	Flexible Macroblock Ordering
FPGA	Field-Programmable Gate Array
IBC	International Broadcasting Convention
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU - Telecommunication Standardization Sector
JTC	(ISO/IEC) Joint Technical Committee
JVT	(MPEG/VCEG) Joint Video Team
HLP	(H.263++) High Latency Profile
MP@ML	(MPEG-2) Main Profile at Main Level
MPEG	(ISO/IEC) Moving Picture Experts Group
NAL	Network Abstraction Layer
PAL	Phase Alternation Line
PSNR	Peak Signal-to-Noise Ratio
QAM	Quadrature Amplitude Modulation
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
QPSK	Quadrature (Quaternary) Phase-Shift Keying
SRAM	Static Random Access Memory
UMTS	Universal Mobile Telecommunication System
VCEG	(ITU-T) Video Coding Experts Group
VCL	Video Coding Layer
xDSL	*(Different variants of)* Digital Subscriber Line

chroma components. All luma and chroma samples of a macroblock are either spatially or temporally predicted, and the resulting prediction residual is transmitted using transform coding. Therefore, each colour component of the prediction residual is subdivided into blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and transmitted using entropy-coding methods.

The macroblocks are organized in slices, which generally represent subsets of a given picture that can be decoded independently. The transmission order of macroblocks in the bitstream depends on the so-called *Macroblock Allocation Map* and is not necessarily in raster-scan order. H.264/AVC supports five different slice-coding types. The simplest one is the **I** slice (where "I" stands for intra). In I slices, all macroblocks are coded without referring to other pictures within the video sequence. On the other hand, prior-coded images can be used to form a prediction signal for macroblocks of the predictive-coded **P** and **B** slices (where "P" stands for predictive and "B" stands for bi-predictive).

The remaining two slice types are **SP** (switching P) and **SI** (switching I), which are specified for efficient switching between bitstreams coded at various bit-rates. The Inter prediction signals of the bitstreams for one selected SP frame are quantized in the transform domain, forcing them into a coarser range of amplitudes. This coarser range of amplitudes permits a low bit-rate coding of the difference signal between the bitstreams. SI frames are specified to achieve a perfect match for SP frames in cases where Inter prediction cannot be used because of transmission errors.

In order to provide efficient methods for concealment in error-prone channels with low delay applications, a feature called *Flexible Macroblock Ordering* (FMO) is supported by H.264/AVC. FMO specifies a pattern that assigns the macroblocks in a picture to one or several slice groups. Each slice group is transmitted separately. If a slice group is lost, the samples in spatially neighbouring macroblocks that belong to other correctly-received slice groups can be used for efficient error concealment. The allowed patterns range from rectangular patterns to regular scattered patterns, such as chess boards, or to completely random scatter patterns.

Intra-frame prediction

Each macroblock can be transmitted in one of several coding types depending on the slice-coding type. In all slice-coding types, two classes of intra coding types are supported, which are denoted as INTRA-4×4 and INTRA-16×16 in the following. In contrast to previous video coding standards where prediction is conducted in the transform domain, prediction in H.264/AVC is always conducted in the spatial domain by referring to neighbouring samples of already coded blocks.

When using the INTRA-4×4 mode, each 4×4 block of the luma component utilizes one of nine prediction modes. Beside DC prediction, eight directional prediction modes are specified. When utilizing the INTRA-16×16 mode, which is well suited for smooth image areas, a uniform prediction is performed for the whole luma component of a macroblock. Four prediction modes are supported. The chroma samples of a macroblock are always predicted using a similar prediction technique as for the luma component in Intra-16x16 macroblocks. Intra prediction across slice boundaries is not allowed in order to keep all slices independent of each other.

Motion compensation in P slices

In addition to the Intra macroblock coding types, various predictive or motion-compensated coding types are specified for P-slice macroblocks. Each P-type macroblock corresponds to a specific partitioning of the macroblock into fixed-size blocks used for motion description. Partitions with luma block sizes of 16×16, 16×8, 8×16 and 8×8 samples are supported by the syntax corresponding to the Inter-16×16, Inter-16×8, Inter-8×16 and Inter-8×8 P macroblock types, respectively. In cases where the Inter-8×8 macroblock mode is chosen, one additional syntax element for each 8×8 sub-macroblock is transmitted. This syntax element specifies if 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. *Fig. 2* illustrates the partitioning.

The prediction signal for each predictive-coded m×n luma block is obtained by displacing an area of the corresponding reference picture, which is specified by a translational motion vector and a picture reference index. Thus, if the macroblock is coded using the Inter-8x8 macroblock type, and each sub-macroblock is coded using the Inter-4x4 sub-macroblock type, a maximum of sixteen motion vectors may be transmitted for a single P-slice macroblock.

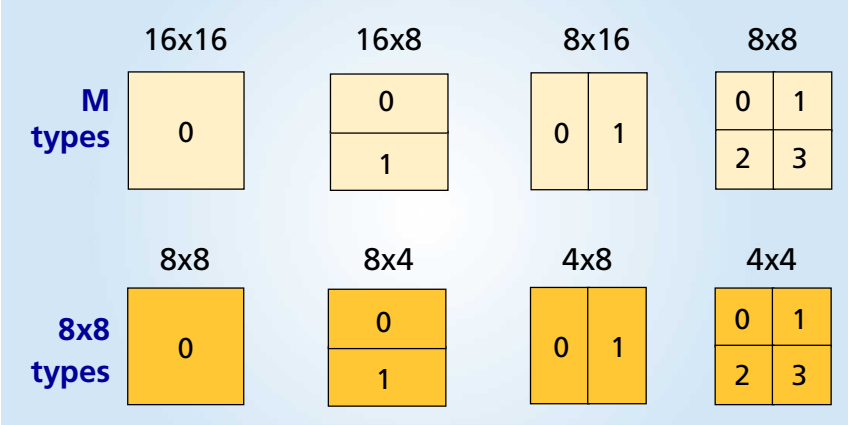

Figure 2
Segmentations of the macroblock for motion compensation.
***Top:* segmentation of macroblocks.**
***Bottom:* segmentation of 8x8 partitions.**

The accuracy of motion compensation is a quarter of a sample distance. In cases where the motion vector points to an integer-sample position, the prediction signals are the corresponding samples of the reference picture; otherwise, they are obtained by using interpolation at the sub-sample positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter. Prediction values at quarter-sample positions are generated by averaging samples at the integer- and half-sample positions.

The prediction values for the chroma components are always obtained by bi-linear interpolation.

The H.264/AVC syntax generally allows unrestricted motion vectors, i.e. motion vectors can point outside the image area. In this case, the reference frame is extended beyond the image boundaries by repeating the edge pixels before interpolation. The motion vector components are differentially coded using either median or directional prediction from neighbouring blocks. No motion vector component prediction takes place across slice boundaries.

H.264/AVC supports multi-picture motion-compensated prediction. That is, more than one prior-coded picture can be used as a reference for motion-compensated prediction. *Fig. 3* illustrates the concept.

Both the encoder and decoder have to store the reference pictures used for Inter-picture prediction in a multi-picture buffer. The decoder replicates the multi-picture buffer of the encoder, according to the reference picture buffering type and any memory management control operations that are 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 signalled. The reference index parameter is transmitted for each motion-compensated 16×16, 16×8, 8×16 or 8x8 luma block.

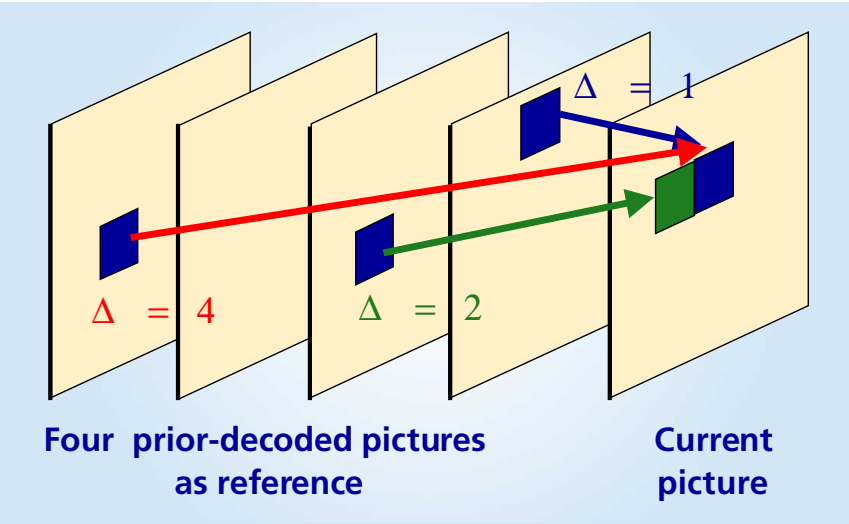

Figure 3
Multi-frame motion compensation. In addition to the motion vector, also picture reference parameters (Δ) are transmitted. The concept is also extended to B pictures as described below.

In addition to the motion-compensated 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, has to be transmitted. The reconstructed

signal is obtained in a similar way to the prediction signal of an Inter-16×16 macroblock that references the picture, which is located at index 0 in the multi-picture buffer. In general, the motion vector used for reconstructing the SKIP macroblock is identical to the motion vector predictor for the 16×16 block. However, if special conditions hold, a zero motion vector is used instead.

Motion compensation in B slices

In comparison to prior video-coding standards, the concept of B slices is generalized in H.264/AVC. For example, other pictures can reference B pictures for motion-compensated prediction, depending on the memory management control operation of the multi-picture buffering. Thus, the substantial difference between B and P slices 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. Generally, B slices utilize two distinct reference picture buffers, which are referred to as the *first* and *second* reference picture buffer, respectively. Which pictures are actually located in each reference picture buffer is an issue for the multi-picture buffer control, and an operation very similar to the well-known MPEG-2 B pictures can be enabled.

In B slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. While list 0 prediction indicates that the prediction signal is formed by utilizing motion compensation from a picture of the first reference picture buffer, a picture of the second reference picture buffer is used for building the prediction signal if list 1 prediction is used. In the bi-predictive mode, the prediction signal is formed by a weighted average of a motion-compensated list 0 and list 1 prediction signal. The direct prediction mode is inferred from previously transmitted syntax elements and can be either list 0 or list 1 prediction or bi-predictive.

B slices utilize a similar macroblock partitioning to P slices. Besides the Inter-16×16, Inter-16×8, Inter-8×16, Inter-8×8 and the Intra modes, a macroblock type that utilizes direct prediction, i.e. the direct mode, is provided. Additionally, for each 16×16, 16×8, 8×16, and 8×8 partition, the prediction method (list 0, list 1, bi-predictive) can be chosen separately. An 8×8 partition of a B-slice macroblock can also be coded in direct mode. If no prediction error signal is transmitted for a direct macroblock mode, it is also referred to as *B slice SKIP mode* and can be coded very efficiently, similar to the SKIP mode in P slices. The motion vector coding is similar to that of P slices with the appropriate modifications because neighbouring blocks may be coded using different prediction modes.

Transform, scaling and quantization

Similar to previous video coding standards, H.264/AVC also utilizes transform coding of the prediction residual. However, in H.264/AVC, the transformation is applied to 4×4 blocks, and instead of a 4×4 discrete cosine transform (DCT), a separable integer transform – with basically the same properties as a 4×4 DCT – is used. Since the inverse transform is defined by exact integer operations, inverse-transform mismatches are avoided. An additional 2×2 transform is applied to the four DC coefficients of each chroma component. If a macroblock is coded in Intra-16x16 mode, a similar 4x4 transform is performed for the 4x4 DC coefficients of the luma signal. The cascading of block transforms is equivalent to an extension of the length of the transform functions.

For the quantization of transform coefficients, H.264/AVC uses scalar quantization. One of 52 quantizers is selected for each macroblock by the Quantization Parameter (QP). The quantizers are arranged so that there is an increase of approximately 12.5% in the quantization step size when incrementing the QP by one. The quantized transform coefficients of a block are generally scanned in a zigzag fashion and transmitted using entropy coding methods. For blocks that are part of a macroblock coded in field mode, an alternative scanning pattern is used. The 2×2 DC coefficients of the chroma component are scanned in raster-scan order. All transforms in H.264/AVC can be implemented using only additions to, and bit-shifting operations on, the 16-bit integer values.

Entropy coding

In H.264/AVC, two methods of entropy coding are supported. The default entropy coding method uses a single infinite-extend codeword set for all syntax elements, except the quantized transform coefficients. Thus, instead of designing a different VLC table for each syntax element, only the mapping to the single codeword table is customized according to the data statistics. The single codeword table chosen is an exp-Golomb code with very simple and regular decoding properties.

For transmitting the quantized transform coefficients, a more sophisticated method called Context-Adaptive Variable Length Coding (CAVLC) is employed. In this scheme, VLC tables for various syntax elements are switched, depending on already-transmitted syntax elements. Since the VLC tables are well designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using just a single VLC table.

The efficiency of entropy coding can be improved further if Context-Adaptive Binary Arithmetic Coding (CABAC) is used [3]. On the one hand, the use of arithmetic coding allows the assignment of a non-integer number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities much greater than 0.5. On the other hand, the use of adaptive codes permits adaptation to non-stationary symbol statistics. Another important property of CABAC is its context modelling. The statistics of already-coded syntax elements are used to estimate the conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods, using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit-rate of between 10 - 15% when coding TV signals at the same quality.

In-loop deblocking filter

One particular characteristic of block-based coding is visible block structures. Block edges are typically reconstructed with less accuracy than interior pixels and "blocking" is generally considered to be one of the most visible artefacts with the present compression methods. For this reason H.264/AVC defines an adaptive in-loop deblocking filter, where the strength of filtering is controlled by the values of several syntax elements. 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 with typically 5-10% while producing the same objective quality as the non-filtered video.

Fig. 4 illustrates the performance of the deblocking filter.

Figure 4
Performance of the deblocking filter for highly compressed pictures.
***Left:* without the deblocking filter. *Right:* with the deblocking filter.**

Interlace coding tools

Frames can be coded as one unit or can be split into two fields which can be coded as separate units again. This field coding is especially efficient if the first field is coded using I slices and the second field makes a prediction from it using motion compensation. Furthermore, field coding is often utilized when the scene shows strong horizontal motion.

In some scenarios, parts of the frame are more efficiently coded in field mode while other parts are more efficiently coded in frame mode. Hence, H.264/AVC supports macroblock-adaptive switching between frame and field coding. For that, a pair of vertically connected macroblocks is coded as two frame or field macroblocks. The prediction processes and prediction residual coding is then either conducted assuming a frame, or field to be coded. The deblocking filtering takes place for all macroblock pairs when they are put into the frame in frame mode, regardless of whether they have been coded in frame or field mode.

Profiles and levels

Profiles and levels specify the conformance points. These conformance points are designed to facilitate interoperability between various applications of the H.262/AVC standard that have similar functional requirements. A *profile* defines a set of coding tools or algorithms that can be used in generating a compliant bitstream, whereas a *level* places constraints on certain key parameters of the bitstream.

All decoders conforming to a specific profile have to support all features in that profile. Encoders are not required to make use of any particular set of features supported in a profile but have to provide conforming bitstreams. In H.264/AVC, three profiles are defined – Baseline, Main and X:

- The **Baseline** profile supports all features in H.264/AVC except the following two feature sets:
  - **Set 1**: B slices, weighted prediction, CABAC, field coding and macroblock adaptive switching between frame and field coding.
  - **Set 2**: SP and SI slices.
- The first set of features is supported by **Main** profile. However, Main profile does not support the FMO feature which is supported by the Baseline profile.
- **Profile X** supports both sets of features on top of the Baseline profile, except for CABAC and macroblock adaptive switching between frame and field coding.

In H.264/AVC, the same set of level definitions is used with all profiles, but individual implementations may support a different level for each supported profile. Eleven levels are defined, specifying upper limits for the picture size (in macroblocks), the decoder-processing rate (in macroblocks per second), the size of the multipicture buffers, the video bit-rate and the video buffer size.

Comparison of H.264/AVC coding efficiency with that of prior coding standards

For demonstrating the coding performance of H.264/AVC [2], we compared it to the successful prior coding standards MPEG-2 Visual [1], H.263++ [3], and MPEG-4 Visual [4] for a set of popular QCIF (10 Hz and 15 Hz) and CIF (15 Hz and 30 Hz) sequences with different motion and spatial detail information. The QCIF sequences were: *Foreman*, *News*, *Container Ship* and *Tempete*. The CIF sequences were: *Bus*, *Flower Garden*, *Mobile and Calendar* and *Tempete*. Based on [5][6], all video encoders were optimized with regards to their rate-distortion efficiency using Lagrangian techniques. In addition to the performance gains, the use of a unique and efficient coder control for all video encoders allowed a fair comparison between them in terms of coding efficiency.

During these tests, the MPEG-2 Visual encoder generated bitstreams at the well-known MP@ML conformance point, and the H.263++ encoder used the features of the High Latency Profile (HLP). In the case of

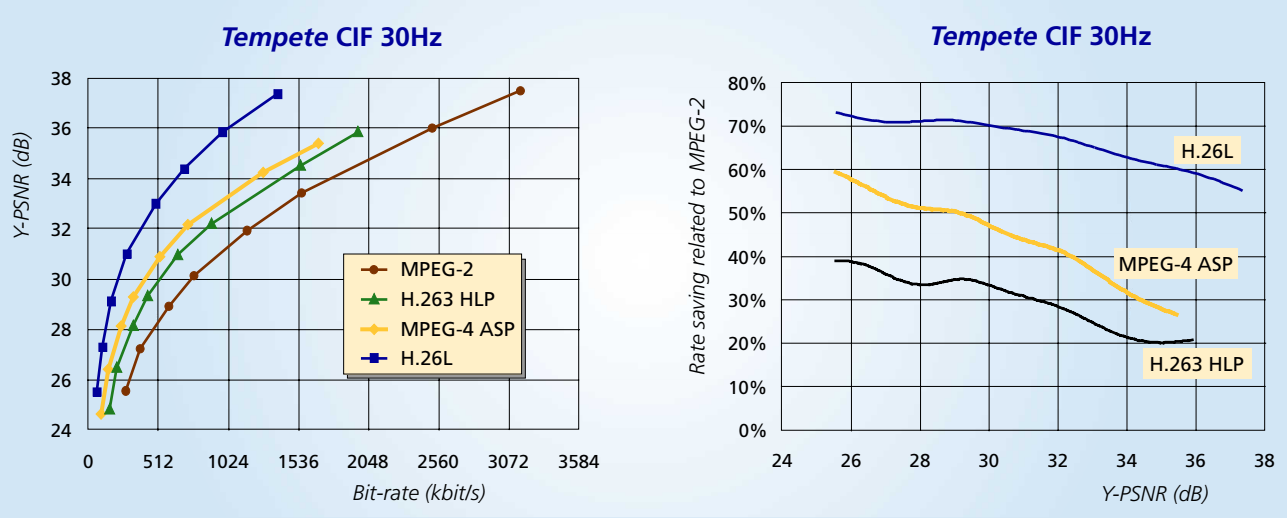

Figure 5
Selected rate-distortion curves and bit-rate saving plots

MPEG-4 Visual, the Advanced Simple Profile (ASP) was used with quarter-sample-accurate motion compensation and global motion compensation enabled. Additionally, the recommended deblocking/deringing filter was applied as a post-processing operation.

For the H.264/AVC JM-2.0 coder, the features enabled in the Main profile were used. We generally used five reference frames for both H.263 and H.264/AVC, with the exception of the News sequences where we used more reference frames for exploiting the known redundancies within this special sequence. With all the coders under test, only the first picture of each sequence was coded as an I-picture, and two B-pictures were inserted between two successive P-pictures. For H.264/AVC, the B-pictures were not stored in the multi-picture buffer, and thus the following pictures did not reference them. Full search motion estimation, with a range of 32 integer pixels, was used by all the encoders along with the Lagrangian coder control from [5][6]. The bit-rates were adjusted by using a fixed quantization parameter.

Fig. 5 shows the rate-distortion curves of all four codecs, for the sequence *Tempete* in CIF resolution.

On the right-hand chart in *Fig. 5*, the bit-rate saving relative to the worst tested video coding standard, MPEG-2, is plotted against the PSNR of the luma component for H.263 HLP, MPEG-2 ASP and H.264/AVC (marked as H.26L). The average bit-rate savings provided by each encoder, relative to all other tested encoders over the entire set of sequences and bit-rates, are depicted in *Table 1*. It can be seen that H.264/AVC significantly outperforms all other standards. The highly flexible motion model and the very efficient context-based arithmetic-coding scheme are the two primary factors that enable the superior rate-distortion performance of H.264/AVC.

Table 1
Average bit-rate savings compared with various prior decoding schemes

Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPEG-4 ASP	-	16.65%	42.95%
H.263 HLP	-	-	30.61%

Although not discussed in this article, the bit-rates for TV or HD video (at broadcast and DVD quality) are reduced by a factor of between 2.25 and 2.5 – when using H.264/AVC coding.

New application areas and business models

The increased compression efficiency of H.264/AVC offers new application areas and business opportunities. It is now possible, to transmit video signals at about 1 Mbit/s with TV (PAL) quality, which enables streaming over xDSL connections. Another interesting business area is TV transmission over satellite. By choosing 8-PSK and turbo coding (as currently under discussion for DVB-S2) and the usage of H.264/AVC, the number of programmes per satellite can be tripled in comparison to the current DVB-S systems using MPEG-2. Given this huge amount of additional transmission capacity, even the exchange of existing set-top boxes might become an interesting option.

Also for DVB-T, H.264/AVC is an interesting option. Assuming the transmission parameters which have been selected for Germany (8k mode, 16-QAM, code rate 2/3, and ¼ Guard Interval), a bitrate of 13.27 Mbit/s is available in each 8 MHz channel. Using MPEG-2 coding, the number of TV programmes per channel is restricted to four whereas, by using H.264/AVC, the number of programmes could be raised to ten or even more, because not only the coding efficiency but also the statistical multiplex gain for variable bit-rates is higher due to the higher number of different programmes. Another interesting option, relating to the discussions on "electro-smog", is to use QPSK, code rate ½ in conjunction with H.264/AVC. This combination would allow us to retain four programmes per channel, but to decrease the transmitted power by 15% in comparison to the transmission mode mentioned above (16 QAM, 2/3).

A further interesting business area is HD transmission and storage. It now becomes possible to encode HD signals at about 8 Mbit/s which fit onto a conventional DVD. This will surely stimulate and accelerate the home cinema market, because it is no longer necessary to wait for the more expensive and unreliable blue DVD laser. It is also possible to transmit 4 HD programs per satellite or cable channel, which makes this service much more attractive to broadcasters, as the transmission costs are much lower than with MPEG-2.

Also in the field of mobile communication, H.264/AVC will play an important role because the compression efficiency will be doubled in comparison to the coding schemes currently specified by 3GPP for streaming [7], i.e. H.263 Baseline, H.263+ and MPEG-4 Simple Profile. This is extremely important because the data rate available in 3G systems works out to be very expensive.

Implementation reports

The H.264/AVC standard only specifies the decoder, as this has been the usual procedure for all other international video coding standards before. Therefore, the rate-distortion performance and complexity of the encoder is up to the manufacturers. Nevertheless, the JVT always requests – for every decoder feature that is proposed – an example encoding method that demonstrates the feasibility of usage of that feature, together with the associated benefits. If the feature is adopted, the proponent is requested to integrate it into the reference software. During the development of H.264/AVC, about 100 proposals from 20 different companies have been integrated into the reference software, making this piece of software very slow and not usable for practical implementation. Therefore, complexity analysis – based on the reference software, e.g., as reported in [8] – typically overstates the actual complexity of the H.264/AVC encoder (by an order of magnitude) and that of the decoder (by a factor of 2 - 3).

In September 2002, at IBC in Amsterdam, VideoLocus showed a demo consisting of its own highly-optimized H.264/AVC codec, running a DVD-quality video stream at 1 Mbits/s in a side-by-side comparison with an MPEG-2 video stream at 5 Mbits/s. VideoLocus' encoder algorithms run on a Pentium 4 platform with hardware acceleration coming from an add-in FPGA card which performs motion estimation, estimation of Intra-prediction, mode decision statistics and video-preprocessing support [9].

In October 2002, UBVideo [10] showed (for the H.264/AVC Baseline profile) CIF-resolution video running on a 800 MHz Pentium 3 laptop computer. The encoding was at 49 frames per second (fps), decoding at 105 fps, and encoding and decoding together at 33 fps. Their low-complexity encoding solution – which is designed/optimized for real-time conversational video applications – incurred an increase in bit-rate of approximately 10% against the rate-distortion performance of the very slow reference software, when encoding typical video content used in such applications.

Like many other companies including Deutsche Telekom, Broadcom, Nokia or Motorola, the Heinrich Hertz Institute (in Berlin, Germany) is developing H.264/AVC real-time solutions. A software implementation, running on a Pentium 4 platform, achieves real-time TV-resolution decoding and 20 Hz CIF encoding with less than 10 - 15 % bit-rate increase over the rate-distortion performance of the very slow reference software. HHI's decoder implementation has been ported on an ARM922 processor, running at 200 MHz, SRAM, showing 6 fps video at CIF resolution and 25 fps video at QCIF resolution.

Conclusions

H.264/AVC represents a major step forward in the development of video coding standards. It typically outperforms all existing standards by a factor of two and especially in comparison to MPEG-2, which is the basis for digital TV systems worldwide; an improvement factor of 2.25 - 2.5 has been reached. This improvement enables new applications and business opportunities to be developed. Example uses for DVB-T, DVB-S2, DVD, xDSL and 3G have been presented. Although H.264/AVC is 2 -3 times more complex than MPEG-2 at the decoder and 4 - 5 times more complex at the encoder, it is relatively less complex than MPEG-2 was at its outset, due to the huge progress in technology which has been made since then.

Another important fact is that H.264/AVC is a public and open standard. Every manufacturer can build encoders and decoders in a competitive market. This will bring prices down quickly, making this technology affordable to everybody. There is no dependency on proprietary formats, as on the Internet today, which is of utmost importance for the broadcast community.

Bibliography

[1] ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2): **Generic coding of moving pictures and associated audio information – Part 2: Video**
ITU-T and ISO/IEC JTC1, November 1994.

[2] T. Wiegand: **Joint Final Committee Draft**
Doc. JVT-E146d37ncm, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), November 2002.

[3] ITU-T Recommandation H.263: **Video coding for low bit-rate communication**
Version 1, November 1995; Version 2 (H.263+), January 1998; Version 3 (H.263++), November 2000.

[4] ISO/IEC 14496-2: **Coding of audio-visual objects – Part 2: Visual**.
ISO/IEC JTC1. MPEG-4 Visual version 1, April 1999; Amendment 1 (Version 2), February 2000.

[5] T. Wiegand and B.D. Andrews: **An Improved H.263 Coder Using Rate-Distortion Optimization**
ITU-T/SG16/Q15-D-13, April 1998, Tampere, Finnland.

[6] G.J. Sullivan and T. Wiegand: **Rate-Distortion Optimization for Video Compression**
IEEE Signal Processing Magazine, Vol. 15, November 1998, pp. 74 - 90.

[7] 3GPP TS 26.233 version 5.0.0 Release 5: **End-to-end transparent streaming service; General description**
Universal Mobile Telecommunications System (UMTS), March 2002.

[8] M. Ravasi, M. Mattavelli and C. Clerc: **A Computational Complexity Comparison of MPEG4 and JVT Codecs**
Doc. JVT-D153r1-L, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), July 2002, Klagenfurt, Austria.

[9] VideoLocus Inc.: **AVC Real-Time SD Encoder Demo, July 2002**
Doc. JVT-D023, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), July 2002, Klagenfurt, Austria.

Ralf Schäfer received his Dipl.-Ing. and Dr-Ing. degrees (both in electrical engineering) from the Technical University of Berlin in 1977 and 1984 respectively. In October 1977, he joined the Heinrich-Hertz-Institut (HHI) in Berlin and, since 1989, he has been head of the Image Processing Department where he is responsible for 55 researchers and technicians, about 40 students and about 25 R&D projects. The main R&D fields are Image Processing, Image Coding, Multimedia Communication over (wireless) Internet, Immersive Telepresence Systems and RT-SW implementations and HW design including VLSI.

Dr Schäfer has participated in several European research activities and was chairman of the Task Force on "Digital Terrestrial Television - System Aspects" of the DVB project, which specified the DVB-T standard. Currently, he is a member of the German "Society for Information Technology" (ITG) where he is chairman of the experts committee "TV Technology and Electronic Media" (FA 3.1) and chairman of the experts group "Digital Coding" (FG 3.1.2).

Thomas Wiegand is head of the Image Communication Group in the Image Processing Department of the Heinrich Hertz Institute in Berlin, Germany. 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 Erlangen-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, USA, 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) standardization efforts. From 1997 to 1998, he has been a Visiting Researcher at Stanford University, USA, and served as a consultant to 8x8 (now Netergy Networks), Inc., Santa Clara, CA, USA.

In October 2000, Dr Wiegand was appointed as Associated Rapporteur of the ITU-T Video Coding Experts Group. In December 2001, he was appointed as Associated Rapporteur / Co-Chair of the Joint Video Team (JVT) that has been created by the ITU-T Video Coding Experts Group and the ISO Moving Pictures Experts Group (ISO/IEC JTC1/SC29/WG11) for finalization of the H.264/AVC video coding standard. He is also the editor of H.264/AVC. His research interests include video compression, communication and signal processing as well as vision and computer graphics.

Heiko Schwarz is with the Image Processing Department of the Heinrich Hertz Institute in Berlin, Germany. He received a Dipl.-Ing. degree from the University of Rostock in 1996 and a Dr-Ing. degree from the University of Rostock in 2000. In 1999, he joined the Heinrich Hertz Institute in Berlin. His research interests include image and video compression, video communication as well as signal processing.

[10] A. Joch, J. In and F. Kossentini: **Demonstration of "FCD-Conformant" Baseline Real-Time Codec**, Doc. JVT-E136, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), October 2002, Geneva, Switzerland.

Acknowledgements

The authors would like to thank Anthony Joch and Faouzi Kossentini for generating and providing the MPEG-2 and H.263+ test results.