

# The H.264 Video Compression Standard

Till Halbach

Department of Telecommunications

Norwegian University of Science and Technology (NTNU)

Trondheim, Norway

Email: [halbach@tele.ntnu.no](mailto:halbach@tele.ntnu.no)

**Abstract**—The recently ratified new video compression standard H.264/MPEG-4 AVC is reviewed. Its basic concepts as well as all important features/algorithms are explained in detail, it is put into the historical context of other video coding standards, and an objective and subjective performance evaluation is given. Compared to previous standards, the coding improvements of H.264 are, in terms of *PSNR*, at least 2 dB and, measured in bit rate saving, at least 40%.

## I. INTRODUCTION

A new international video compression standard has seen the light of the day recently. ITU-T Recommendation H.264 [1] and ISO/IEC International Standard 14496-10 (MPEG-4 AVC – Advanced Video Coding – or MPEG-4 Part 10) [2] are identical and the result of standardization efforts since 1998. The project, more commonly known under the label H.26L, was started by the Video Coding Experts Group (VCEG) of ITU-T to carry on the development in video compression from the former standard H.263++. In December 2001, VCEG merged with Moving Pictures Experts Group (MPEG) of ISO/IEC to form the Joint Video Team in order to bundle the standardization activities and develop the standard on a joint basis. H.264 was ratified as recommendation by ITU-T in May 2003, and MPEG-4 AVC became international standard in June the same year. The standard is the first third-generation video coding scheme after the first generation with H.120, H.261 and MPEG-1, and the second generation which consists of H.263, MPEG-2 and MPEG-4. There is no backward compatibility of H.264 to the former H.263++ and MPEG-2 Video (Part 2).

The article is organized as follows. The standardized algorithms are divided into basic features and those which are included in specialized feature sets. After introduction and detailed explanation of these algorithms, the article turns to the standard's performance, compared to existing popular video coding standards. Conclusions and an outlook sum up at the end.

## II. BASIC TECHNICAL CONCEPTS

The new standard addresses – like other specifications in the H.26x and MPEG-x series – decoder issues only. That is, it describes the coded picture buffer (CPB) which contains the compressed data, the decoding engine, and the decoded picture buffer (DPB), compound of the reconstructed pictures, see Fig. 1. The interaction of these components is specified by the timing model of the so-called hypothetical reference decoder.

In addition to that, syntax and semantics of the compressed data, i.e. code stream elements, are described. Encoder matters and decoder issues like error concealment are mentioned as informal supplements and are not an integral part of the standard. The normative part includes the core specifications and Annexes A through E with additional specifications which are mandatory for a standard-conform decoder.



Fig. 1

A GENERIC DECODER AS DEFINED IN THE SPECIFICATIONS

For a better illustration of the block-based hybrid coding scheme of H.264, the block diagram of an H.264-conform encoder is derived from the specifications, see Fig. 2. A spatial or temporal prediction of the current signal is computed and subtracted from the original. This prediction error is then transformed, quantized and entropy-encoded. The standard requires a binary input signal representation and generates in turn binary channel symbols, i.e. a bit stream.

To reduce complexity requirements, H.264 works on so-called macroblocks (MBs) which consist of one  $16 \times 16$ -pixel luminance (luma) block and – assuming a YCbCr color space and 4:2:0 chrominance (chroma) subsampling of the input signal – two blocks of  $8 \times 8$  pixels for the color components. The blocking artifacts caused by the MB approach are reduced by an adaptive in-loop antiblocking filter which applies non-linear filtering to all block edges. The filtering process reduces the bit rate typically by 5-10% at the same visual fidelity.

H.264 consists like its predecessors of several sets of algorithms, also called profiles, of which one has to be chosen for a specific application. Each algorithm incorporated by a particular set contributes to a small extend to the overall impressive savings in favor of H.264 as compared to previous standards. The specification of a particular profile is complemented by a so-called level which defines limitations on parameter values like picture size and processing rate.

So far, there are three profiles. Baseline is a low-complexity low-latency profile which provides basic functionality. Typical application are interactive ones like mobile video telephony and video conferencing. The Main profile targets studio, broadcast, and other high-quality applications like HDTV and

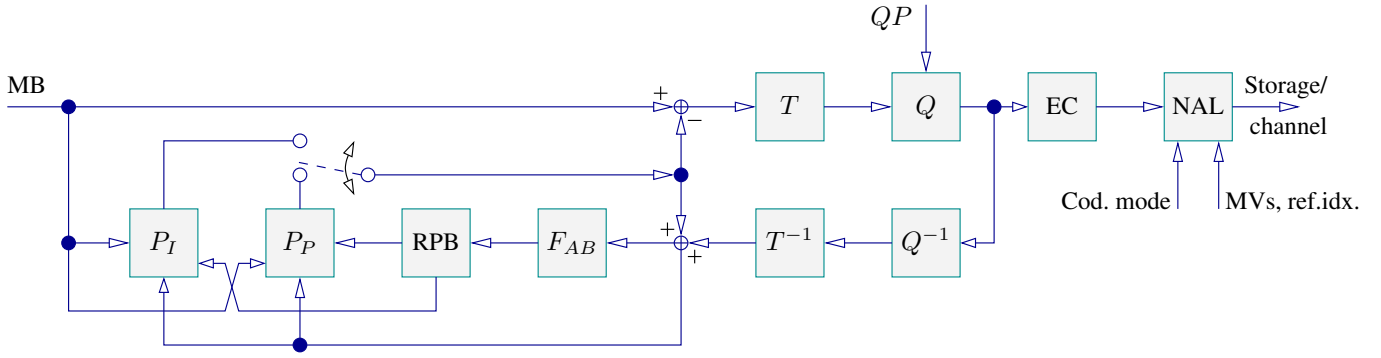


Fig. 2

AN ENCODER WHICH PRODUCES A STANDARD-CONFORM BIT STREAM. THE REFERENCE PICTURE BUFFER (RPB) IS IDENTICAL WITH DPB.  $P_I$  DENOTES INTRA AND  $P_P$  INTER PREDICTION. ANTIBLOCKING FILTER, TRANSFORM, INVERSE TRANSFORM, QUANTIZATION AND INVERSE QUANTIZATION ARE REPRESENTED BY  $F_{AB}$ ,  $T$ ,  $T^{-1}$ ,  $Q$ ,  $Q^{-1}$ , RESPECTIVELY. THE EC BLOCK PERFORMS ENTROPY CODING AND PASSES THE RESULTING CODE STREAM TO THE NETWORK ABSTRACTION LAYER. THERE, THE ENCODED DATA IS MULTIPLEXED WITH SIDE INFORMATION LIKE CODING MODE, MOTION VECTORS AND REFERENCE INDICES, AND FINALLY TRANSMITTED. ALL BLOCKS EXCEPT FOR THE NAL AMOUNT TO THE VIDEO CODING LAYER (VCL)

DVD. It is a high-complexity high-latency profile without error-resilient features. Finally, Extended is highly error-robust and of low complexity and may be used in e.g. streaming applications. All profiles have Baseline compatibility.

Spatial prediction, also known as INTRA or I coding, is based on the samples of the current picture. There are various directional modes, of which one is illustrated in Fig. 3. Depending on the samples available for reference, H.264 supports nine regular modes with  $4 \times 4$  luminance blocks, four modes with  $16 \times 16$  luminance blocks to improve coding of large uniform luminance areas, and four modes with  $8 \times 8$  chrominance blocks. The  $4 \times 4$  prediction modes are differentially coded and then, together with the other modes, passed to the NAL.

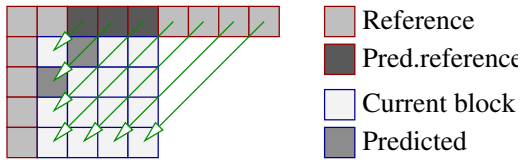


Fig. 3

SPATIAL PREDICTION OF LUMINANCE SAMPLES WITH DESIGNATION 'INTRA  $4 \times 4$  DIAGONAL DOWN LEFT'

The signal's temporal prediction, also referred to as INTER or P coding, is made from samples belonging to a previously decoded picture. Motion estimation/compensation (ME/MC) operates on blocks of variable size to adapt precisely to the motion within an image sequence. MBs and  $8 \times 8$ -pixel blocks can be divided into subblocks with one or both dimensions cut into halves as depicted in Fig. 4. This leads to a hierarchical tree-structured motion segmentation with possible block sizes  $(x \times y)$   $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$ , and  $4 \times 4$ . All subblocks within one MB must be of the same type (I/P).

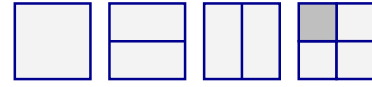


Fig. 4

POSSIBLE BLOCK SPLITS OF  $16 \times 16$ - AND  $8 \times 8$ -PIXEL BLOCKS. THE GRAY-SHADED BLOCK IS THUS EITHER OF SIZE  $8 \times 8$  OR  $4 \times 4$  PIXELS

The quarter-pel accuracy of the ME/MC process is achieved by the combination of a six-tap filter with the coefficients (1, -5, 20, 20, -5, 1) and a two-tap filter defined by (1, 1). Along image boundaries, the samples are extrapolated. The temporal prediction of the chrominance signal employs bilinear interpolation and a re-use of the luminance motion vectors. All motion vectors are differentially encoded by means of a median estimate or, in the case of  $8 \times 16$ - and  $16 \times 8$ -pixel blocks, a direct estimate. The reference samples can be taken from several frames, which is also named the concept of multiple reference frames. This concept necessitates the conveyance of the index to the reference frame for each block down to the size  $8 \times 8$ . It is stressed that the maximum number of motion vectors per MB can be 16.

The main transform in H.264 is a separable DCT-approximating  $4 \times 4$  integer transform, which has the advantage that problems like coefficient drifting and encoder/decoder mismatch can be avoided. Associated with the transform is an appropriate scaling later in the quantization stage. The transform matrix  $T_4$  is given by

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}. \quad (1)$$

It is – independently of the block mode – applied to the prediction error first horizontally and then vertically. There is an

additional  $2 \times 2$  integer transform available for the chrominance components and – as a second step – an additional  $4 \times 4$  integer transform applied to the first-step DCT coefficients in large uniform luminance areas, i.e.  $16 \times 16$  blocks. All transforms are low-complexity transforms which can be realized without multiplication and using 16-bit arithmetic.

The transformed coefficients in the encoder are quantized by means of one quantizer out of the set of 52 uniform scalar quantizers. The quantizer's step size is controlled by a quantization parameter  $QP$ . The  $QPs$  are defined such that the quantization factor doubles with an increase of the parameter by the value six and offer a wide range of possible image qualities. There is no weighted quantization because no gain could be shown for this technique so far due to the small transform size.

Most side information and header data is encoded by a single variable-length code table, an exponential Golomb code with parameter zero. There is no need for table storage as the code is regular with a variable-length prefix of the form  $00\dots01$  and a fixed-length suffix. The first code word entries of the table are listed in Tab. I. Some syntax elements have in advance to be mapped to the indices of code words due to their probability distribution, such that often occurring symbols are assigned small code word indices, which in turns results in short code words. The transform coefficients are treated differently, as explained in Sec. II-A and Sec. II-B.

Index	Code word
0	1
1	010
2	011
3	00100
4	00101
$\vdots$	$\vdots$

TABLE I

FIRST ENTRIES IN THE EXPONENTIAL GOLOMB CODE TABLE

#### A. Baseline profile

The Baseline profile makes use of all algorithms described so far. Additionally, low-complexity entropy coding of the quantized transform coefficients is specified, so-called context-adaptive variable-length coding (CAVLC). CAVLC exploits the coefficients' statistical correlation by first scanning them in a zig-zag manner into a one-dimensional array. Every non-zero coefficient is then associated with a variable *run* which counts the number of zero coefficients to the previous non-zero coefficient.

It can be observed that there are very often 1's with either sign among the highest-frequency coefficients. These are recorded in number (up to three) and, together with the total number of non-zero coefficients, coded with one out of a set of code tables. The decision which table to use is made with regard to the number of non-zero coefficients in neighboring blocks. Additionally, the sign of the 1's has to be conveyed to the decoder. The values of the remaining coefficients is

then coded using adaptive Rice codes, where the adaptivity is given by a varying suffix size to adapt to the coefficients' frequency range. That is, several code tables are used, and the choice among the tables is made according to the value of the previously encoded coefficient. After that, the sum of run's is computed and encoded with one out of 15 tables depending upon the number of non-zero coefficients in that block. Now, the only thing that remains is to code the individual run values with one out of seven code tables, depending upon the remaining sum of run's. All code tables used by CAVLC have been generated empirically.

Consider as an example the 1-D array of coefficients (12, -7, 0, 0, 5, 1, -1, 0, -1, 1, 0,  $\dots$ , 0) from which the sequence of non-zero coefficients (12, -7, 5, 1, -1, -1, 1) and the associated run's (0, 0, 2, 0, 0, 1, 0) are extracted. The number of non-zero coefficients is 7, and the number of trailing 1's is 3, leading to the 2-tuple (7,3) which is variable-length coded. The signs of these coefficients is (+, -, -), each signaled with one bit to the decoder. The sum of run's is  $2 + 1 = 3$ . Finally, the remaining coefficients (1, 5, -7, 12) and the respective run's (0, 1, 0, 0, 2, 0, 0) are encoded in a backward manner. The decoder determines the position of the highest-frequency coefficient by adding the number of non-zero coefficient and the sum of runs, assuming that array indexing starts with one. The trailing 1's and other coefficients can hereafter be placed according to the successively decoded individual run's.

In H.264, one or several MBs are grouped into a structure called slice group. A slice group is in turn composed of one or several slices. It is hereby possible to limit the size of slices according to the packet length requirements of a given network. Each MB belongs to exactly one slice which in turn may only consist of MBs of one picture. A slice is the smallest independently decodable unit in an encoded video. As such, any prediction referring to a location beyond slice boundaries is not allowed, and e.g. the respective samples and motion vectors are marked as not available. The use of slices stops spatial error propagation and reduces simultaneously the coding efficiency. An I slice may contain only I MBs, and a P slice may be compound of both I and P MBs.

H.264 knows four different slice group (*SG*) allocation types, also known as flexible MB ordering (FMO). First, there is either a horizontal or vertical raster scan of MBs (Fig. 5(a) and Fig. 5(b), respectively), i.e. filling the slice group row-/column-wise. A combination of both are rectangular slices which consist of contiguous areas of MBs. If carefully designed, rectangular slices (Fig. 5(c)) do not reduce the coding efficiency very much but, nevertheless, bound the error impact to only a limited spatial area. Then, dispersive slices (Fig. 5(d)) are tailored for heavily interference-prone channels. Concentrated transmission errors are spread over the whole spatial plane and may efficiently be concealed by the decoder. However, this scheme reduces prediction gains significantly. If one of these concepts should not suffice, MBs can explicitly be allocated to a slice group, one by one.

There are three other error resilience techniques specified in the Baseline profile. Redundant slices (RS) allow the insertion

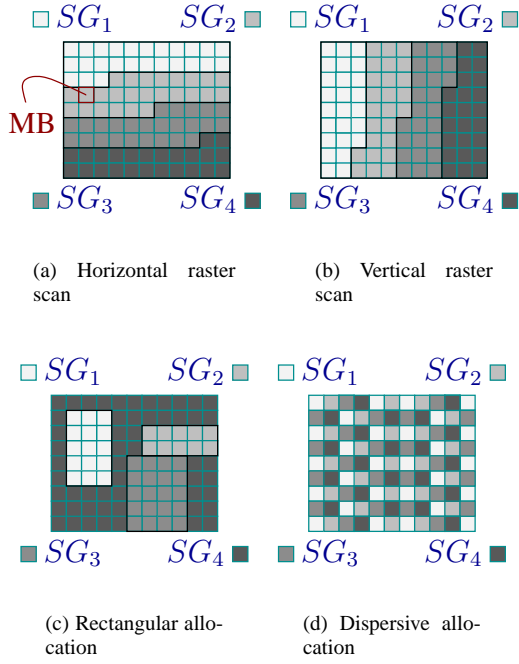


Fig. 5

ALLOCATION TYPES OF MBs TO SGs. FOR SIMPLICITY OF ILLUSTRATION, THE NUMBER OF SLICES PER SLICE GROUP IS SET EQUAL TO ONE

of primary and secondary slices in the bit stream. If a primary slice is affected by errors, it can be replaced by an error-free redundant one, otherwise the redundant slices are discarded. This feature is also useful in e.g. a simulcast environment where the primary slices are coded with a high and the other slices with a low bit rate. Also, in order to account for e.g. IP-type networks, the decoder allows the slices to arrive in arbitrary order (ASO). And finally, messages containing supplemental enhancement information (SEI) may contain further information about the bit stream, which can be utilized e.g. by an error concealment scheme.

### B. Main profile

In the Main profile, all Baseline tools excluding ASO, FMO, and RS are supported. The transform coefficients are coded by so-called context-adaptive binary arithmetic coding (CABAC). Its block diagram is shown in Fig. 6. CABAC is more complex than CAVLC but outperforms it by typically 5–15% in terms of bit rate savings.

The syntax elements (source symbols) are first binarized either by plain fixed- or variable-length coding, the latter using the regular code shown in Tab. II. There are roughly 400 non-hidden contexts defined in H.264, of which one is chosen for a specific element. A context represents knowledge about the element to code, like its meaning (e.g. motion vector) and the values of previously encoded elements associated with the same context. Each context determines a probability estimation

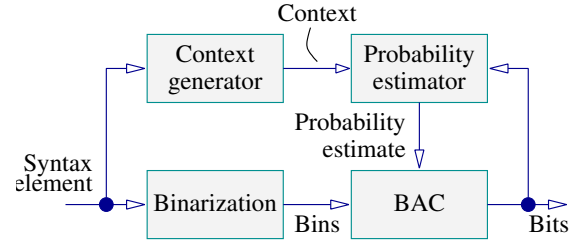


Fig. 6

ARCHITECTURE OF CABAC. THE CORE ENGINE IS THE BINARY ARITHMETIC CODER (BAC)

which is in turn feed into the arithmetic coder. The estimates are updated after each coding of a bin by means of a finite-state machine. At the beginning of a slice, the states are reset to their initial values. The arithmetic coder has been developed exclusively for H.264 and operates very close at the entropy limit.

Index	Code word
0	1
1	01
2	001
⋮	⋮

TABLE II

BINARIZATION TABLE

Another feature which increases the complexity considerably in the Main profile is the use of B slices which may contain I, P, and B MBs. In contrast to P blocks which refer to only a single picture, B blocks make use of two reference pictures, hereby mean averaging two predictions to the total estimate. The referencing may be carried out in a temporally forward and backward manner, however, also double forward and double backward predictions are possible. Unlike previous standards, B pictures can be marked as reference pictures.

The Main profile offers the use of weighted predictions, i.e. linear transformation of one or two predictions in the form  $X_p = a \cdot X_r + c$  with P pictures and  $X_p = a \cdot X_{r,1} + b \cdot X_{r,2} + c$  with B pictures, where  $X_p$  denotes the prediction and  $X_r$  the reference signal. This is especially useful for changes in intensity like e.g. fades. Each reference picture can be associated with a separate weighting.

The last presented feature in this section is the Main profile's ability to process video by MB-adaptive frame/field (MBAFF) coding, as illustrated in Fig. 7. It may be referred to the same frame (but a different field) for temporal prediction. In frame mode, a picture is identical with a frame. In field mode, a picture is identical with either top or bottom field. The results of MBAFF are improved coding gain when coding interlaced sources and a better position for error concealment as discussed in e.g. [3].

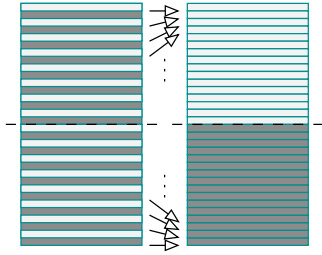


Fig. 7

MB-ADAPTIVE FRAME/FIELD CODING. THE VERTICALLY CONNECTED MB PAIR ON THE LEFT SIDE CAN BE CODED AS TWO SEPARATE MBs WITH SAMPLE LINES OF ALTERNATING FIELDS (TOP AND BOTTOM), OR AS TWO MBs (TOP AND BOTTOM) WITH SAMPLE LINES OF UNIFORM FIELD AFFILIATION

### C. Extended profile

Like in Main, Extended allows partly the MBAFF feature, depending on the chosen level. Further, there are two main tools specified, both increase the standard's error robustness. Data partitioning (DP) allows to group the encoded slice data into three partitions according to the classification header, INTRA-specific, and INTER-specific data. Each partition is in turn transmitted as one packet by the NAL. DP eases the use of UEP schemes that account for the non-uniform importance of the data as a reference for subsequent prediction.

Finally, Extended allows the use of so-called switching I and P slices, or short SI/SP. This technique aims at stream switching, stream skipping, error resilience, etc. The philosophy beyond S slices is that a reconstructed S slice is always identical with e.g. an I, P, or B slice of another stream, such that decoding of the stream can continue even though a different reference, i.e. the S slice, is used. SI slices are self-contained without any temporal prediction, and SP slices correspond to P slices.

## III. PERFORMANCE EVALUATION

Preliminary tests to assess the compression performance of H.264 have shown promising results. The intra-frame coding abilities were compared to the performances of the still-image coding standards JPEG, JPEG2000, and the upcoming Motion JPEG2000 in [4]. More than 5 dB gain on the average in contrast to JPEG and more than 1 dB in contrast to JPEG2000 were reported. Considering the overall rate distortion behavior, interim tests have shown that the emerging standard (TML-8) outputs only half the bit rate of an MPEG-4 codec (Advanced Simple profile) for the same visual fidelity [5]. Operating at the same rate, the average image *PSNR* for H.26L-coded material is typically 2–4 dB better than for MPEG-4. H.26L's gain to H.253++ (Advanced Simple profile) is even higher: Between 2.5 and 5 dB have to be expected. Compared to MPEG-2, H.26L performs approximately 6 dB better on the average.

As the standard's development has been finished by May/June 2003, a final assessment of its compression performance

is in place. Subsequently, the performance of H.264 is compared to the previous successful coding standards MPEG-2 Visual [6], H.263++ [7], and MPEG-4 Visual [8] for error-free transmission. An evaluation of the standard's error robustness can be accessed in e.g. [11]. All encoders utilize rate distortion optimization as suggested in [9] to achieve fair testing conditions.

In the following, only the representative results of one video out of the set of CIF-size test videos with an encoded frame rate of 30 fps are shown (Fig. 8). All 300 frames of the video are encoded in frame coding mode. Only the initial frame is coded as I frame. The reference software version JM-7.3, operated at Main profile, was employed for all obtained results. The number of reference frames is set to five, and the search range is 32 pixel wide. Two consecutive B frames are inserted between P frames. The target rate is controlled by passing various *QP* values in the range 25–45 to the encoder.

It can be observed that H.264 outperforms all competitors significantly over the whole rate spectrum. The gain in *PSNR* to the closest opponent MPEG-4 is largest (roughly 2 dB) at moderate to high rates and decreases somewhat for low rates (around 128 Kbps). The curves correspond to bit rate savings of H.264 with respect to MPEG-4 of approximately 40% with the same visual fidelity, a result consistent with [10].

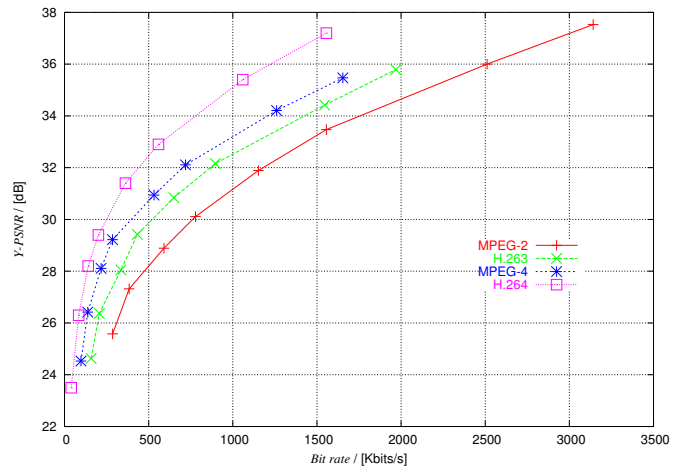


Fig. 8

THE DISTORTION RATE CURVES OF ALL CODECS WITH THE *Tempete* VIDEO. THE DR CURVES FOR MPEG-2 VISUAL, H.263++, AND MPEG-4 VISUAL ARE TAKEN FROM THE LITERATURE [10]

About 90% of the rate is on the average used for coding the luma signal, 10% is associated with the chroma signal. Considering different slice types, it is noticed that the number of bits spent for coding P frames is 10% of the amount used up on I slices, whereas B slices are responsible for 2% of the rate of the I frame consumption. Generally, the rate used on side information is increased in H.264 compared to previous standards.



#### IV. CONCLUSIONS AND OUTLOOK

An overview of the recently released video compression standard H.264 was given. The author hopes that the article eases reading of the more than 250 pages of standard specifications. The overview included a sketch of the standardization efforts and the historical context as well as an introduction to the standard's basic functionality. This was followed by a detailed explanation of the three algorithm sets specified so far. The article concluded with a performance evaluation with respect to existing video coding standards.

It can be confirmed that the H.26L project has reached its goals set forward. H.264's quality increase in contrast to the named previous standards in terms of *PSNR* is excellent. There are several reasons for that. First, there is a number of new coding tools standardized in the specifications. Second, already known algorithms succeed in a better adaptation to the original video signal by means of small block sizes down to  $4 \times 4$  pixels. The very small transform block sizes witness about the system's good adaptation to source statistics by effective removal of temporal and spatial redundancy. Furthermore, H.264 is a flexible standard which gives much freedom to the implementor. An example for this is the fact that also B pictures may be used as reference for temporal prediction – of course depending on the desired application.

The standardized algorithms have an increased computational complexity and require more memory than previous standards. (In [10], the average decoder is claimed to be 2–3 times more complex than most MPEG-2 decoders. For the encoder, the range of 4–5 applies.) However, it was the intention of the new specification to account for the availability of cheap memory and the progress made in the area of powerful processing units. The industry is obviously willing and/or able to pay the price.

The block-based hybrid coding scheme proves once more to be successful. Bit rates from below 64 Kbps up to more than 100 Mbps can be achieved, corresponding to image material of the sizes sub-QCIF and ITU-R 601 up to sizes appropriate for HDTV and DVD. The list of potential applications is immense as the new standard has been developed for both conversational/real-time and non-conversational services, starting with the classic video conferencing and video phone, and also including terrestrial television broadcasting as well as direct

broadcast satellite video services. Especially to mention are applications utilizing H.32x series packet transmission and, RTP/UDP/IP-type communication. Its excellent compression efficiency will H.264 further open new markets like digital cinema and storage of medical imagery. The royalty-free Baseline profile will surely contribute to this development. Due to the wide-spread coalition of standardization participants, H.264 is expected to replace previous standards like H.263++ quite soon, and MPEG-4 AVC is likely to be used in place of MPEG-2 Visual for its target applications. Real-time solutions of 16-bit implementations have already been presented, and new products are ready to enter the markets.

As the L in H.26L stands for long-term (development), it is expected that the standardization be continued as with H.263 formerly. New profiles and levels are for sure to come, and further annexes may extend the existing ones in the near future. A draft of the standard can be found on <ftp://ftp.itmct-files.org/jvt-experts>. For useful news about MPEG-4, see <http://www.m4if.org>.

#### REFERENCES

- [1] *Video Coding for Very Low Bit Rate Communication*, ITU-T Recommendation H.264, May 2003.
- [2] *Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding*, ISO/IEC International Standard 14496-10, June 2003.
- [3] T. Halbach and T. A. Ramstad, "Multidimensional adaptive non-linear filters for concealment of interlaced video," in *Proc. Nordic Signal Processing Symposium (NORSIG)*, Bergen (Norway), Oct. 2003.
- [4] T. Halbach and M. Wien, "Concepts and performance of next-generation video compression standardization," in *Proc. Nordic Signal Processing Symposium (NORSIG)*, on board Hurtigruten (Norway), Oct. 2002.
- [5] P. Topiwala, G. Sullivan, A. Joch, and F. Kossentini, "Performance evaluation of H.26L, TML 8 vs. H.263++ and MPEG-4," ITU-T Q.6/SG 16 (VCEG), Tech. Rep. N18, Sept. 2001.
- [6] *Generic coding of moving pictures and associated audio information – Part 2: Video*, ISO/IEC International Standard 13818-2, Nov. 1994.
- [7] *Video Coding for Low Bitrate Communication*, ITU-T Recommendation H.263v3, Nov. 2000, note.
- [8] *Information Technology – Coding of Audio-Visual Objects – Part 2: Visual*, ISO/IEC International Standard 14496-2, Jan. 2000.
- [9] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.
- [10] R. Schäfer, T. Wiegand, and H. Schwarz, "The emerging H.264/AVC standard," *EBU Technical Review*, Jan. 2003.
- [11] T. Halbach and S. Olsen, "Error robustness evaluation of H.264/MPEG-4 AVC," accepted at VCIP, Visual Communications and Image Processing (Jan. 2004).