to appear in JOURNAL OF VLSI SIGNAL PROCESSING

Performance of the H.263 Video Compression Standard

Bernd Girod, Niko Färber, Eckehard Steinbach

Telecommunications Institute
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
{girod, faerber, steinb}@nt.e-technik.uni-erlangen.de

Invited Paper

Abstract

The draft international standard ITU-T H.263 is closely related to the well known and widely used ITU-T Recommendation H.261. However, H.263 does provide the same subjective image quality at less than half the bit-rate. In this paper we investigate to what extend single enhancements of H.263 contribute to this performance gain, and consider the trade-off quality vs. complexity. Based on typical test sequences, H.263 with its various modes is compared to H.261 on the basis of rate distortion curves at bit-rates up to 128 kbps. At 64 kbps, the performance gain of H.263 in its default mode compared to H.261 is approximately 2 dB. This improvement is achieved with only little increase of complexity, and is mainly due to more accurate motion compensation with half-pel accuracy. Considering the trade-off quality vs. complexity, the combination of the optional coding-modes "Advanced prediction mode" and "PB-frames mode" is a good compromise, resulting in an additional performance gain of 1.5 dB PSNR at 64 kbps. The "Syntax-based arithmetic coding mode" on the other hand, offers only a very small performance gain (0.1 dB at 64 kbps) for its increased computational complexity. Results from profiling an H.263 software codec are presented in order to support complexity considerations of the optional coding-modes.

1 Introduction

The demand for videophone applications over Public Switched Telephone Networks (PSTN) or mobile links requires new standards for the compression of image sequences at very low bit-rates. The ITU-T Study Group XV has recently devised a standard suitable for video transmission below 64 kbps.

The ITU-T recommendation H.263 [1] is closely related to the well-known and widely used ITU-T recommendation H.261 [2], which has been devised by the same Study Group. This close relationship helped to arrive at the new standard in a short period of time, including not only the video coding algorithm but also the corresponding audio (G.723), multiplex (H.223), control (H.245) and system (H.324) aspects. Though H.261 and H.263 share the same basic codec structure, there is a significant improvement in performance. Side-by-side comparisons show that the same subjective image quality can be achieved with less than half the bit-rate. This performance gain is due to improved and optimized coding techniques, as well as optional coding-modes ("options") which may be negotiated between coder and decoder. For a given application, however, the trade-off quality vs. complexity has to be considered as well. Therefore, a systematic comparison of the H.263 and H.261 video compression standards regarding quality and complexity is presented in this paper. We assume that the reader is familiar with H.261.

The paper is organized as follows. In section 2 we briefly describe the hybrid coding structure common to H.261 and H.263. In section 3 the differences between the two standards are pointed out, followed by a description of the additional options of H.263 in section 4. Then we compare the performance of H.261 and H.263 with several options in section 5. The increase in complexity due to the H.263 options and real-time implementation issues are discussed in section 6.

2 Motion-Compensated Hybrid Coder

Both H.263 [1] and H.261 [2] share the same basic structure of the encoder (Fig. 1). It is a hybrid of interframe prediction exploiting temporal redundancy and transform coding of the residual prediction error exploiting spatial redundancy and adaptively reducing spatial resolution. Temporal prediction is based on block-based motion estimation (ME) and compensation (MC), while a discrete cosine transform (DCT) is used for spatial redundancy reduction.

The motion compensated prediction error is subdivided into 8×8 blocks and each block is transformed by the DCT. The coefficients are quantized (Q) and pairs of zero-runs and quantizer levels are combined, resulting in a series of variable length codewords (VLC) for each block. One of 31 uniform threshold quantizers may be selected adaptively. Finally, run-level pairs, motion vectors (MV), and quantization parameters (Qp) are entropy coded along with other side information and multiplexed to the bit-stream.

3 Enhancements in Base-line H.263

Though H.261 and H.263 share the same basic coding structure, there are both minor and major differences. These are covered in this section. A special section is dedicated to the H.263-options, which are further enhancements not included in H.261.

Target bit-rate: The target bit-rate of H.261 is $p \times 64$ kbps ($p = 1,2,\ldots,30$) whereas H.263 aims at bit-rates below 64 kbps. Those bit-rates reflect the typical applications envisioned when the standards were designed. In the case of H.261, this was visual telephony over ISDN, whereas H.263 shall enable the same service over PSTN (e.g., using V.34).

Picture formats: H.261 operates on two picture formats, CIF (Common Intermediate Format) and quarter-CIF (QCIF). As the name implies, the spatial resolution of QCIF is one quarter of the CIF-resolution. In addition to CIF and QCIF, H.263 supports three other formats. As can be seen in Table 1, the spatial resolution is further reduced for sub-QCIF, while 4CIF and 16CIF are formats with an increased resolution. Sub-QCIF resolution may be useful for extremely low bit-rates and inexpensive low-end terminals, while 4CIF and 16CIF indicate that the standard may also be used at higher bit-rates or for freeze-frame transmission.

Format	luminance	chrominance
sub-QCIF	128×96	64×48
QCIF	176×144	88×72
CIF	352×288	176×144
4CIF	704×576	352×288
16CIF	1408×1152	704×576

Table 1: Spatial resolution of supported picture formats

Motion compensation accuracy: While H.261 is limited to motion compensation (MC) with integer-pel accuracy, H.263 provides half-pel accuracy. We will see in section 5 that this extension improves performance significantly. The improvement due to half-pel MC is well understood [3] and has already been utilized successfully in ITU-T H.262 (MPEG-2) [4].

Filter in the loop: H.261 utilizes a spatial lowpass filter in the predictor, the "filter in the loop", which can be switched on a macroblock basis. Though there is a significant performance gain due to loop filtering, H.263 does not include such a loop filter. The reason is that the bilinear interpolation used in H.263

for half-pel MC introduces spatial lowpass filtering as a side affect. In addition, one of the H.263 options includes overlapped block MC (see section 4), which has an inherent filtering effect as well.

GOB layer: Both standards use a hierarchical syntax decomposing a sequence into pictures, group of blocks (GOB), and macroblocks (MB). In H.261 the size of a GOB is defined as 11×3 MBs, whereas in H.263 the size depends on the picture format. For sub-QCIF, QCIF and CIF resolution, one GOB consists of one MB-row, while for 4CIF and 16CIF resolution one GOB consists of two MB-rows. The second difference in the GOB layer is that H.263 allows to insert header information optionally. This enables the coder, e.g., to insert extra synchronization words, or to reduce the overhead for applications requiring a lower robustness.

Other: Besides the differences mentioned above H.263 contains several minor improvements compared to H.261, which shall be explained briefly in the following non-exhaustive list. Though each single improvement only makes a small difference, they all add up and contribute to the overall performance.

- **VLC tables:** According to the new syntax, there are several new tables for variable length codes (VLC). Other tables are optimized for better performance. E. g., the events coded in the VLC table for DCT coefficients now consist of triplets (run, level, eob) rather than pairs (run, level).
- **Motion vector prediction:** Motion vectors are coded differentially as in H.261, but with a more sophisticated predictor. Not only the preceding MB is used for prediction, but also MBs in the previous MB-row.
- **Adaption of quantization parameter:** While H.261 allows any quantizer for any MB, H.263 restricts the transitions from MB to MB within a GOB to the two next coarser or finer quantizers. This yields a smaller and more effective VLC table.

- **Macroblock addressing:** For skipped MBs, H.261 uses a MB address (MBA) indicating how many MBs are skipped. In H.263, for every skipped MB a single bit is transmitted.

4 H.263 Options

An H.263-coder may use optional coding techniques ("options") to further improve its performance. Options have to be negotiated with the decoder via external means (for example within ITU-T H.245). There are four options available in H.263, which will be briefly explained in the following. The reader is referred to Annex D, E, F and G of the H.263 recommendation [1] for more detailed information on these special coding techniques.

4.1 Unrestricted motion vector mode (UMV-mode)

In the default prediction mode of H.263, motion vectors must not reference pixels outside the coded picture area. This restriction need not hold if the UMV-mode is switched on. If a motion vector component points outside the coded picture area, the edge pixel referenced by the truncated vector component is used instead. This extension is very useful for small motion vectors where most pixels in a MB still reference pixels inside the coded picture area. Note that for small image formats like sub-QCIF up to 50% of all MBs are located at the border.

4.2 Advanced prediction mode (AP-mode)

This option includes overlapped block motion compensation and the possibility of four motion vectors per macroblock (instead of one). In addition, the maximum range of motion vector components is extended from [-16, 15.5] to [-31.5, 31.5]. If the AP-mode is switched on, it is automatically used in combination with the UMV-mode.

4.2.1 Four motion vectors per macroblock

This additional coding mode may be selected on a macroblock basis. For each 8×8 luminance block separate motion vectors are transmitted resulting in a more accurate motion compensation for the luminance signal. Similar to the default mode, motion vectors are encoded differentially using the median value of three motion vector components from adjacent 8×8 or 16×16 blocks as a predictor. The trade-off between improved motion compensation and additional side information has to be considered as part of the mode decision in the encoder.

For the chrominance signals only one motion vector is used for motion compensation. It is derived by calculating the sum of the four luminance motion vectors followed by a division by 8 to scale the vector for the coarser chrominance sampling grid.

4.2.2 Overlapped block motion compensation

Overlapped block motion compensation (OBMC) is an advanced prediction scheme, where the current frame of a sequence is predicted by motion compensated overlapping blocks from the reference frame, each weighted by a smoothing window. In H.263, OBMC is performed on the basis of 8×8 blocks for the luminance signal only. Each pixel is a weighted sum of three prediction values. In order to obtain the three prediction values, three motion vectors are used: the motion vector of the current luminance block and two out of four *remote* vectors. Remote vectors are the motion vectors of horizontally and vertically directly adjacent 8×8 blocks. These motion vectors do not in general belong to the same GOB. For each pixel, the remote motion vectors of the blocks at the two nearest block borders are used. Fig. 2 shows the position of the remote motion vectors for the current luminance block.

The three selected motion vectors are then used to form the actual prediction value, which is simply a weighted sum of the three referenced luminance values. Depending on the position in the block, different weights are used for pixels referenced by the current and remote motion vectors. The weights are defined in a matrix that reduces the influence of the remote motion vectors as the position approaches the center of the current 8×8 block. At the borders, a smooth transition between 8×8 blocks is achieved due to the increasing influence of the remote motion vectors. This results in a visible reduction of block artifacts produced by block-wise motion compensation. For a detailed description of the theory underlying OBMC refer to [5].

4.3 PB-frames mode (PB-mode)

A PB-frame consists of two pictures coded as one unit. The P-frame is predicted as in baseline H.263 from the previously decoded P-frame, while the prediction in the B-frame is the average of two motion compensated predictions, one from the previous P-frame and one from the current P-frame. Fig. 3 illustrates the positions of P- and B-frames and the prediction directions.

In contrast to MPEG, the information for B-frames is not contained in a separate frame but combined with the P-frame on a macroblock level. Within each MB, the blocks belonging to the P-frame are first transmitted, followed by the blocks of the B-frame. Therefore, the number of blocks per MB is 12 rather than 6. This joint encoding reduces the overhead for picture and macroblock headers, which is important for very low bit-rates.

As illustrated in Fig. 3, not all pixels in the B-frame are predicted bidirectionally. If the backward prediction is formed from pixels outside the currently reconstructed P-macroblock, the prediction is limited to forward prediction from the previous P-frame. Though this constraint reduces coding efficiency, it guarantees that future P-macroblocks are never used for backward prediction. Therefore, the received bit-stream can be decoded sequentially, which simplifies hardware implementation.

Furthermore, H.263 does not allow the independent coding of forward and backward motion vectors like in MPEG. Especially for sequences with complex motion, this further reduces the coding efficiency. The reason for those constraints is the different purpose of B-frames in H.263, which is explained in more detail in section 5.

4.4 Syntax-based arithmetic coding mode (SAC-mode)

In this optional mode of H.263, all variable length coding/decoding operations with Huffman tables are replaced by arithmetic coding/decoding operations. By using syntax-based arithmetic coding, the restriction of a fixed integral number of bits for each symbol is removed.

5 Rate-Distortion Performance

In this section we compare the rate distortion performance of H.263 and H.261. We use the averaged peak signal-to-noise ratio (PSNR) of the luminance component as a distortion measure for a whole sequence, i. e., first we calculate the PSNR for each luminance frame n according to

$$PSNR_n = 10 \log \frac{1}{M} \sum_{i=1}^{M} \frac{255^2}{(o_i - c_i)^2}, \tag{1}$$

where M is the number of samples in a frame, and o_i and c_i are the amplitudes of the original and coded frame respectively. Then the PSNR values for each frame are averaged for N frames in the sequence according to

$$PSNR = \frac{1}{N} \sum_{n=1}^{N} PSNR_n. \tag{2}$$

The rate is expressed in kbps (1000 bit/s) and includes the portion of the two chrominance components. All data presented in the following were obtained for the "Foreman" test sequence in QCIF resolution (frames 0-200). In addition, the main results are also provided for the two test sequences "Mother and Daughter" and "Salesman" which contain less complex motion. All sequences were coded at a fixed frame-rate using fixed quantizers. Note that a bit-rate control strategy would make a fair comparison difficult. This is especially true if the bit-rate control results in a variable frame-rate, where different frames are encoded in different simulations. The mode decision was done according to the common reference models, i.e., RM8 [6] for H.261 and TMN5 [7] for H.263. For integer-pel motion estimation, both refference models specify full search ME with a search range of ± 15 pel. Additionally, TMN5 specifies a half-pel refinement of ± 0.5 pel around the best integer vector. Simulations were carried out using available software codecs [8] [9].

The results are presented in two parts. The first part covers the actual comparison of H.261 and H.263. We investigate to what extend various improvements contribute to the overall performance gain. In this part we either use all options of H.263 or none. In the second part, we evaluate the effectiveness of the H.263 options compared to the default mode of H.263, thus providing a more differentiated view of the performance gain due to single options.

The results from the comparison of H.261 and H.263 are summarized in Fig. 4. At a bit-rate of 64 kbps, the following observations can be made. H.263 w/o options outperforms H.261 by approximately 2 dB. Another 1.5 dB is gained if we use all of the H.263 options (top curve). More than 50% of the maximum performance gain are apparently due to features not included in the H.263 options (see also Table 2). In order to provide an impression of the subjective image quality, Fig. 5 shows details from the test sequence coded with H.261 and H.263.

For further analysis the curve denoted "H.263 w/o options, integer-pel accuracy" demonstrates the importance of half-pel MC in H.263. It shows the performance of H.263 when we restrict the motion vectors to integer-pel accuracy. Note that the same VLC-Table is used and the distribution of the motion vector bits vs. the coefficient bits does not change significantly, i.e. from 11% vs. 74% to 9% vs. 77%. However, a dramatic loss of 3 dB at 64 kbps can be

(PSNR [dB])	foreman	mother and daughter	salesman
H.261	30.02	33.90	34.61
H.263 w/o options	+1.94	+2.16	+2.36
H.263 with all options	+3.28	+3.76	+3.70

Table 2: Comparison of H.261 and H.263 at 64 kbps and 12.5 fps.

observed. It should be noted that this loss is not only due to less accurate MC. When using integer motion vectors only, the inherent filtering effect of bilinear interpolation is lost as well. This is also the reason why H.263 now performs even worse than H.261.

In order to compare H.261 and H.263 under the same conditions with respect to MC and filtering, consider the bottom curve in Fig. 4. It shows the performance of H.261 with the loop filter switched off. In the bottom two curves neither a loop filter nor half-pel MC is used. At 64 kbps, H.263 still outperforms H.261 by 0.5 dB. This small advantage is due to the 'other' minor differences mentioned at the end of section 3.

Advanced prediction mode: In Fig. 6 the performance gain due to the H.263 options 'Advanced Prediction mode' (AP) is illustrated. Because the unrestricted motion vector mode is included in the advanced prediction mode, this option is not investigated separately. At 64 kbps, the AP-mode results in a performance gain of approximately 1 dB. It should be mentioned that 23% of all macroblocks were coded using four motion vectors and that the distribution of the motion vector bits vs. the coefficient bits changed from 11% vs. 74% to 20% vs. 63%. For test sequences with less motion the gain due to the AP-mode is reduced (see Table 3).

Syntax-based arithmetic coding mode: The improvement due to the SAC-mode is very small, approximately 0.1 dB at 64 kbps (Fig. 7). Because SAC is a different (lossless) entropy coding scheme, the PSNR for a given quantizer is uneffected, but fewer bits are produced. In terms of reduced bit-rate, the average gain for inter-coded macroblocks is 3-4%. For intra-coded macroblocks, the gain is higher, on average about 10%.

PB-frames mode: The main purpose of PB-frames in H.263 is to increase the frame-rate without increasing the bit-rate too much. Typically, the frame-rate is doubled when the PB-mode is used. Consider a sequence coded at 6.25 fps using H.263 w/o options (Fig. 8, top curve). Though the quality of single frames is good (33.6 dB at 64 kbps), the low temporal resolution results in jerky motion. However, increasing the frame-rate to 12.5 fps while maintaining the bit-rate at 64 kbps leads to a significant loss of image quality (1.6 dB).

A better compromise between temporal and spatial resolution is possible when the PB-mode of H.263 is used. The dashed curves in Fig. 8 show the performance of the PB-mode at 12.5 fps. Because the quality of P- and B-frames differs significantly, the averaged PSNR according to Eq. 2 is calculated separately. As can be seen, the PSNR for P-frames drops only by 0.6 dB compared to the top curve. Note that the same number of P-frames per second (6.25) are now transmitted with only little loss of quality. With the use of the B-frames, however, the frame-rate is doubled. Though the quality of the B-frames is low, they provide the subjective impression of smooth motion by exploiting spatio-temporal properties of human vision [10]. According to this concept, the fraction of the bit-rate allocated to the B-part of a PB-frame is kept low, on average about 15-20%. In fact, H.263 specifies that a B-macroblock always has to be quantized more coarsely than its corresponding P-macroblock.

Because the quality of the B-frames is intentionally kept lower, the need for effective bidirectional prediction is not as big. Instead, the design of PB-frames focuses on less overhead for the B-part, which is very important for low bit-rates. Hence, PB-frames in H.263 are not meant to work in the same way as e.g. B-frames in MPEG [4] and will actually be less effective. This is because no independent forward and backward motion vectors can be used for the bidirectional prediction. Instead, a small delta-vector is used to obtain the forward and backward motion vectors from a scaled version of the P-vector. Especially for complex motion with big delta-vectors, this reduced flexibility and the constraint on bidirectional prediction described in section 4.3 results in reduced efficiency. Therefore, the search range for delta-vectors should be kept small. Increasing the search range (which is possible within the standard) does not guarantee a better prediction. In fact, due to more side information the performance might decrease. We will refer to this observation when we consider complexity issues in section 6.

In Tab. 3 it can be seen, that the PB-mode is less effective for the "Foreman" test sequence that contains complex motion. Also note that the quality of the P-frames is consistently higher than the quality of the B-frames.

Fig. 9 shows the performance of H.263 when all the options are combined. At a frame-rate of 12.5 fps, the quality of the P-frames increases by 2.3 dB compared to H.263 w/o options. For the B-frames a gain of 0.4 dB can be observed. Because the influence of the SAC-mode is small, most of the performance gain is due to the combination of the AP- and PB-mode.

6 Computational Complexity

So far we investigated the performance gain regardless of the involved complexity. For a given application, however, the trade-off quality vs. complexity has to be considered as well. In this section we try to give some indication of the involved complexity. Similar to the previous section, we first compare H.263 in its default mode with H.261, and then focus on the H.263 options. We con-

(PSNR [dB])		foreman	mother and daughter	salesman
w/o options		31.96	36.06	36.97
AP		+1.11	+0.30	+0.72
SAC		+0.16	+0.10	+0.10
PB	P	+0.97	+1.53	+1.40
	B	-0.79	+0.55	+0.43
AP-PB-SAC	P	+2.27	+2.16	+1.92
	B	+0.40	+1.03	+0.95

Table 3: Comparison of H.263 options at 64 kbps and 12.5 fps.

clude the section with a brief discussion of real-time implementation on personal computers.

6.1 Comparison H.263 vs. H.261

Though the rate distortion performance of H.263 w/o options is significantly better than H.261, the complexity does not differ too much. On one hand, H.263 requires additional complexity for half-pel ME and MC. Compared to integer-pel ME, however, the additional complexity due to the half-pel refinement search and bilinear interpolation is typically small. On the other hand, H.263 does not require any operations for loop filtering, like H.261.

We conclude that H.263 w/o options is an effective alternative to H.261, since a significant performance gain is provided for a minor increase in complexity.

6.2 Complexity of H.263 options

In the following, we try to assess the additional complexity when single options of H.263 are used. Together with the knowledge of the corresponding performance gain, this will be useful to decide which options should be supported by a coder for a given application. In addition to some basic considerations, we provide results from profiling a H.263 software codec on a Sun Workstation (SuperSPARC, 50 MHz, 134 MIPS, 64 MB memory, Solaris 2.3 Operating System). These data should be used with great care, since they depend heavily on the implementation and may not generalize, especially for hardware implementations. The coder used in our simulations [8] is based on TMN5. Besides other issues like rate control and mode decision, TMN5 does also specify, that motion vectors are obtained by full search motion estimation. Because full search motion estimation is computationally intense, the complexity of the coder is dominated by this operation (see gray bars in Fig. 10).

The H.263 software codec has been profiled under the following conditions. We used the "Foreman" test sequence (frames 0-150) and a constant bit-rate of 30 kbps while maintaining a fixed frame-rate of 10 fps. Single options of H.263 or combinations of them were used in six simulations, which are abbreviated as follows:

- NO: No options
- AP: Advanced Prediction mode
- PB: PB-frames mode
- SAC: Syntax-based Arithmetic Coding mode
- AP-PB: Both AP and PB are used
- AP-PB-SAC: AP, PB and SAC are used

The results are summarized in Fig. 10 and are discussed in the following paragraphs. The case "no options" (NO) serves as the baseline in our discussion.

Advanced prediction mode: The first simulation (AP) shows that the performance gain of approximately 1.2 dB (at 64 kbps) has to be paid for by a considerable increase in complexity. Because the coder may either use one motion vector per MB (MV) or four (MV_1-MV_4), all five motion vectors have to be calculated to enable an effective decision. The additional complexity for ME however, strongly depends on the implementation. The referred software coder for example, uses a small refinement search to obtain MV_1-MV_4 from MV, whereas TMN-5 specifies a full search ME for each motion vector.

In addition, extra complexity is introduced by overlapped block motion compensation. A weighted sum of three prediction values has to be computed to yield the final prediction value. Because this has to be done for every pixel, the computational cost is high. In contrast to ME, which is only done at the coder, overlapped motion compensation does also increase the complexity of the decoder.

PB-frames mode: When comparing the PB-frames mode (PB) to the default mode (NO), the computation time for the coder decreases. The reason is the less expensive motion estimation for the blocks of a B-frame. A full ME is done only between two P-frames - every other frame. In the B-frames a smaller search range of typically ± 2 pels is used. As we have discussed in section 5, the limited search range is mainly selected due to performance considerations, not complexity. A bigger search range will not necessarily increase the performance. Note that in our simulations the frame-rate of the PB-mode is the same as in the default mode. If the PB-mode would be used to double the frame-rate, as discussed in section 5, the complexity would increase significantly. For the same frame-rate, however, the reduced complexity is a positive side effect.

Again, the more complex prediction (averaging values from two frames) increases the complexity at the coder and decoder. Because this is done only for every other frame, and it is less complex than overlapped block motion compensation, the increase in computation time at the decoder is less than in the AP-mode.

Another particularly important point to mention for PB-frames is the increase of the delay caused by this option. On the coder side two frames have to be stored in memory and are coded as one unit which causes an extra delay of one frame. Assuming frame skipping to reduce bit-rate, this delay can be as much as 6 or more frames of the original sequence. However, if the PB-mode is used to double the frame-rate, no additional delay is introduced compared to the default mode.

Syntax-based arithmetic coding mode: The improved performance due to syntax-based arithmetic coding (SAC) of approximately 0.1 dB results in an increased complexity for the coder and decoder. Because the consumed time by motion estimation dominates the overall computational complexity at the coder, the relative increase may be neglected. For the decoder, however, the execution time increases noticeably.

Combination of options: The last two bars in Fig. 10 show the combination of several options. The combination of the AP- and PB-mode results in a complexity similar to the default mode. Though the AP-mode requires a more complex motion estimation, this operation is done only between two P-frames. Therefore, if the AP-mode is combined with the PB-mode, the computational complexity for motion estimation is reduced by 50%. Considering the corresponding performance gain, this combination appears to be a good choice. Adding SAC to this combination increases complexity, but the corresponding performance gain is small as discussed in section 5.

6.3 Real-time Implementation of H.263

The video coding standard H.263 is very suitable to provide a low-cost video conferencing capability for personal computers equipped with a high-speed modem, operating, e.g., at 28.8 kbps. While the H.263 codec used for the CPU-time measurements in this section does not run in real-time on a personal computer today, dramatic speed-ups are possible with an optimized implementation. Such an optimization can exploit three principles:

1. The arithmetic logic units (ALU) of typical CPUs provide a wordlength of 32 bits today, while most image compression operations require an accuracy of 8 to 12 bits. Algorithmic modifications that exploit the full word-length of the ALU therefore yield significant speed-ups.

2. Look-up tables in a software implementation are essentially free, if their size does not exceed a few hundred kByte. This is very different from a hardware realization. Hence, frequently used results should be precomputed in a software implementation of the H.263 codec.

3. Many algorithmic short-cuts are possible, that have no effect on the resulting image quality, but lead to dramatic computational savings. A particularly important area for such optimization is the motion estimator, where intelligent schemes can lead to great savings without sacrificing performance.

A detailed discussion of implementation optimization is beyond the scope of this paper. Software-only video conferencing systems utilizing H.261 have been commercially available since 1994. The system with the by far best demonstrated performance encodes up to 15 fps in QCIF while simultaneously decoding up to 30 fps in full CIF on a 75 MHz Pentium processor [11]. Similar performance has been demonstrated for H.263.

7 Conclusions

In this paper, we have presented a systematic evaluation of the international video coding standard ITU-T H.263. The enhancements of H.263 over the well known and widely used ITU-T Recommendation H.261 have been presented. Mandatory modifications and optional coding-modes have been evaluated with respect to image quality and complexity. It has been shown that half-pel motion compensation is one of the main advantages of the new standard. Therefore, H.263 provides a better image quality than H.261, even when it is used in its default mode. Furthermore, the complexity of H.263 in this configuration is only marginally higher than the complexity of H.261.

Using the AP-mode of H.263 does increase the execution time of the coder and decoder. For the PB-mode, however, a reduced complexity at the coder side can be observed due to less expensive motion estimation in the B-frames. At the decoder the complexity is increased, but by a smaller amount than in the AP-mode. The corresponding performance gain and the special role of B-frames in H.263 have been discussed. Considering the trade-off quality vs. complexity, the combination of AP- and PB-mode appears to be a good compromise. Syntax-based arithmetic coding, on the other hand, offers only little performance gain for its increased computational complexity.

References

[1] "Video Coding for narrow telecommunication channels at $<$ 64 kbit/s", ITU-T Recommendation H.263, 1996.

[2] "Video Codec for audiovisual services at p x 64 kbits/s", ITU-T Recommendation H.261, Geneva, 1990.

[3] B. Girod, "Motion-Compensating Prediction with Fractional-Pel Accuracy", IEEE Transactions on Communications, vol. 41, no. 4, April 1993,

pp. 604-612.

[4] ISO/IEC 13818 (ITU-T H.262), "Generic coding of moving pictures and associated audio information, Part 1: Systems, Part 2: Video, Part 3: Audio", March 1994.

[5] M. T. Orchard, G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach", IEEE Transactions on Image Processing, vol. 3, no. 5, September 1994, pp. 693-699.

[6] "Description of Reference Model 8 (RM8)", CCITT SGXV WG4, Specialists Group on Coding for Visual Telephony, Doc. 525, June 1989.

[7] "Video Codec Test Model, TMN5", ITU-T SGXV WP 15/1, Expert's Group on Very Low Bitrate Visual Telephony, Doc. LBC-95, January 1995.

[8] TMN-Test model for H.263, Version 1.4a Telenor Research, 1995 (Internet: http://www.nta.no/brukere/DVC/).

[9] H.261 software codec, Portable Video Research Group, Stanford (Internet: ftp://havefun.stanford.edu/pub/p64/).

[10] U. Götz and R. Schäfer, "Considerations on the Possibility to Exchange Temporal Against Spatial Resolution in Image Coding", Signal Processing: Image Communication, vol. 2, 1990, pp. 39-51.

[11] Vivo Software, Inc., Waltham, Mass., USA, personal communication.

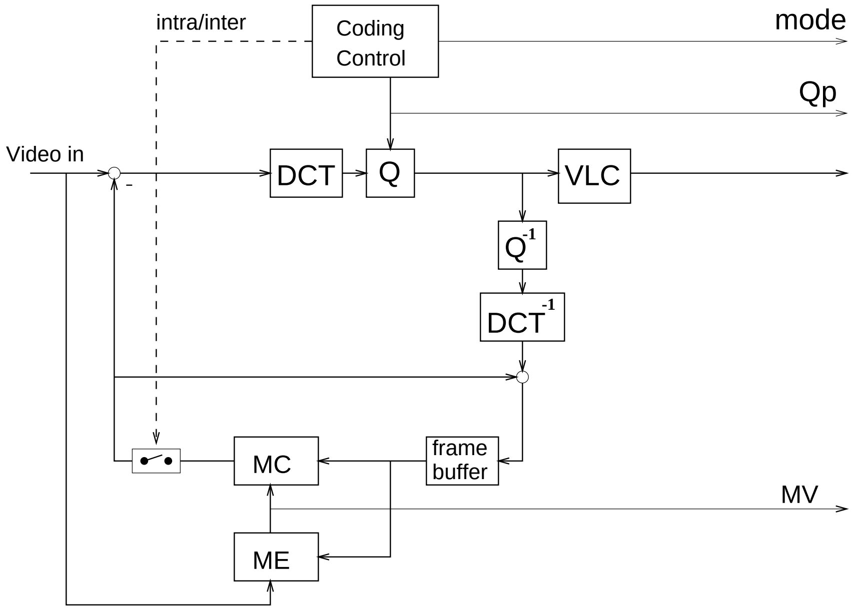

Figure 1: The hybrid coding structure of H.261 and H.263

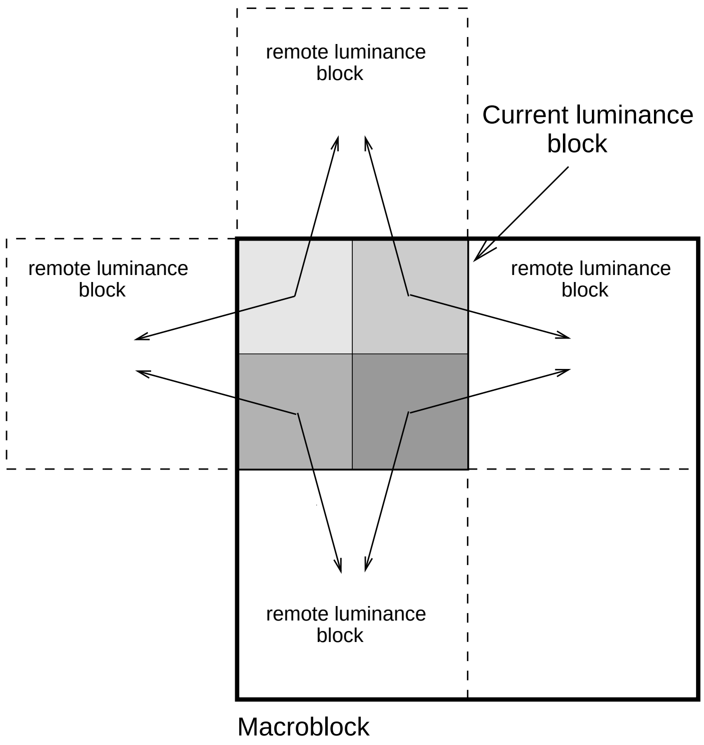

Figure 2: Selection of the remote motion vectors

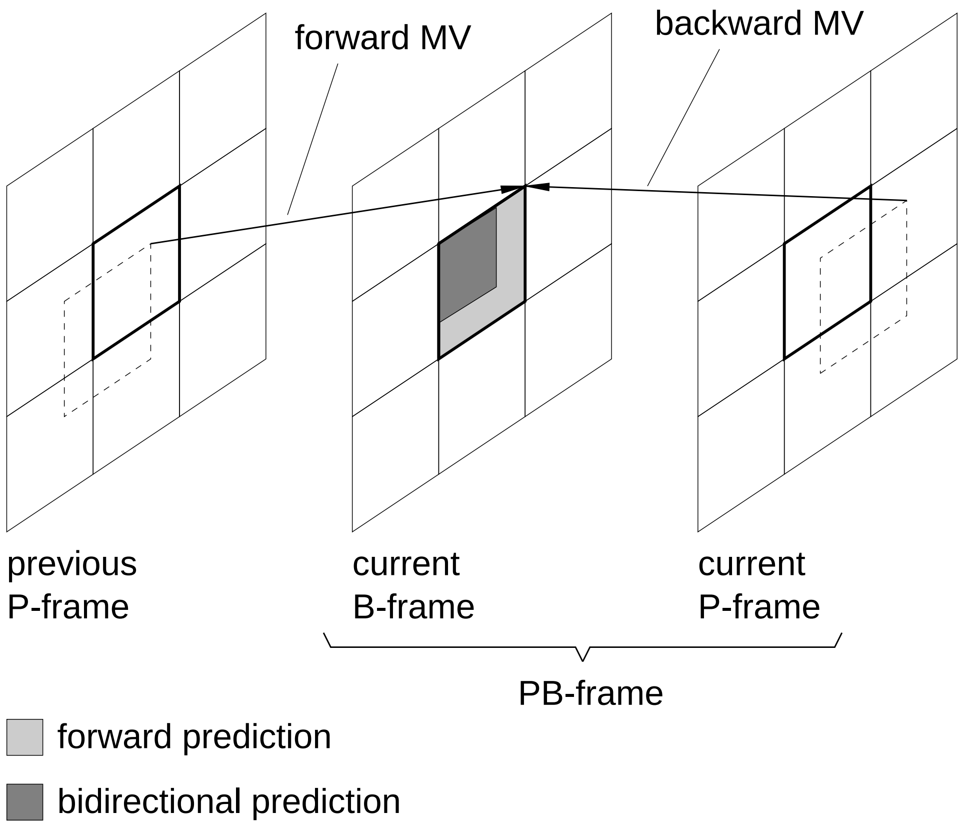

Figure 3: Prediction in PB-frames mode

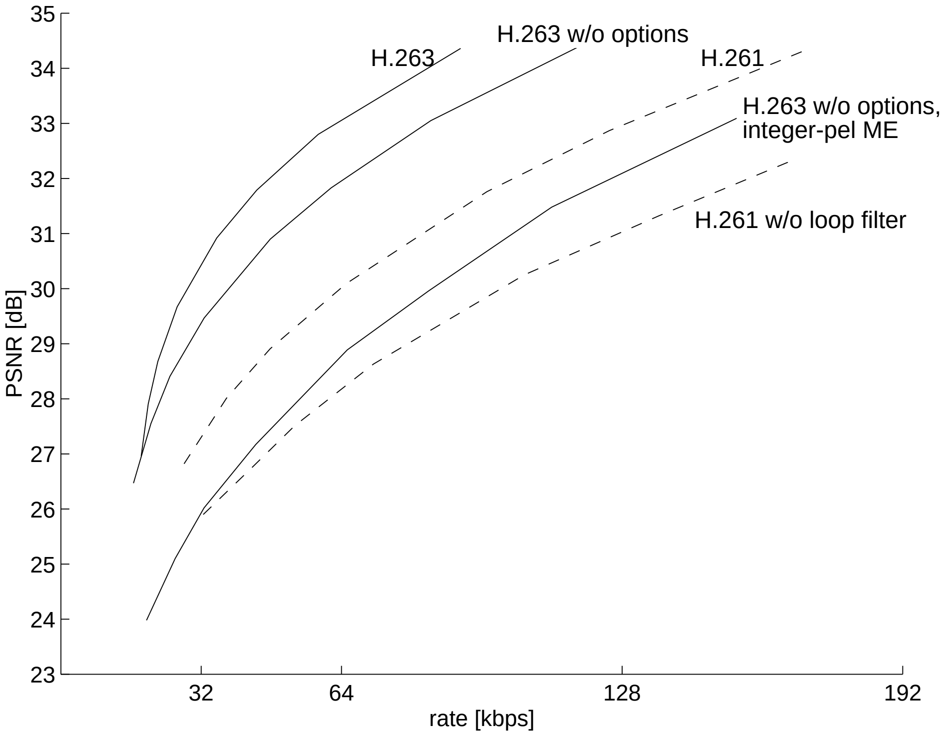

Figure 4: Performance of H.261 and H.263 at a frame-rate of 12.5 fps (Foreman).

Original, 88×72 pel

H.261, PSNR=30.03 dB

H.263 w/o options, PSNR=31.87 dB

H.263 AP-mode, PSNR=32.78 dB

Figure 5: Details from test sequence Foreman coded with H.261 and H.263 (frame-rate 12.5 fps, bit-rate 64 kbps).

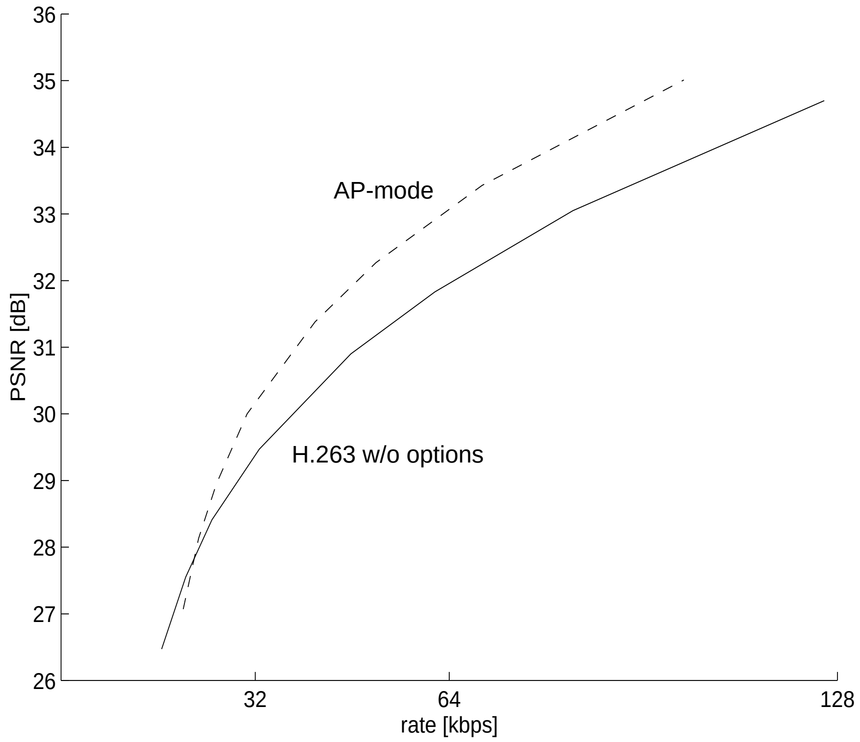

Figure 6: Performance of the H.263 option 'Advanced Prediction mode' at a frame-rate of 12.5 fps (Foreman).

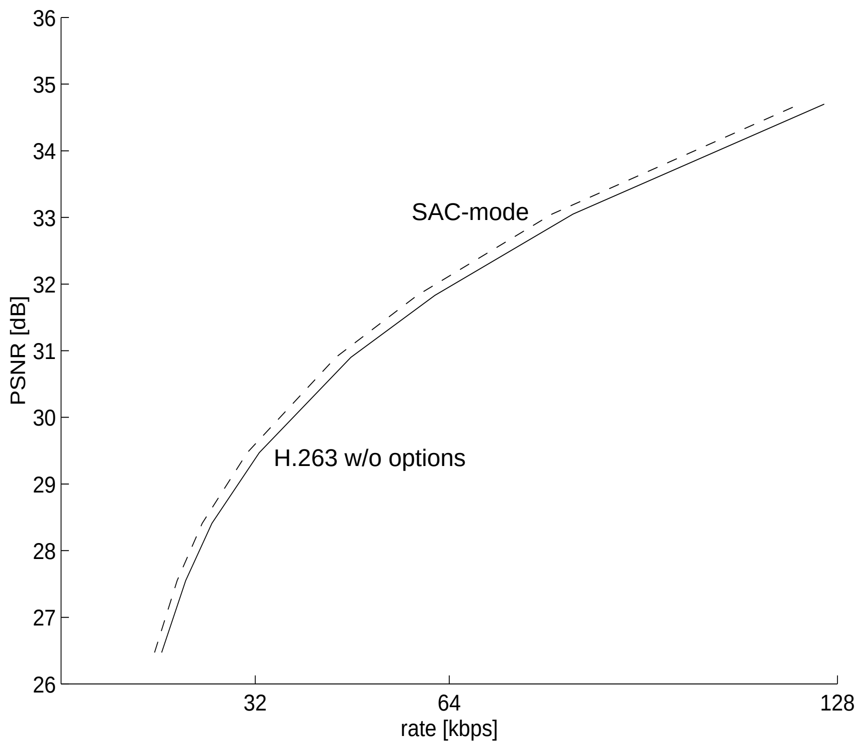

Figure 7: Performance of the H.263 option 'Syntax-based Arithmetic Coding' at a frame-rate of 12.5 fps (Foreman).

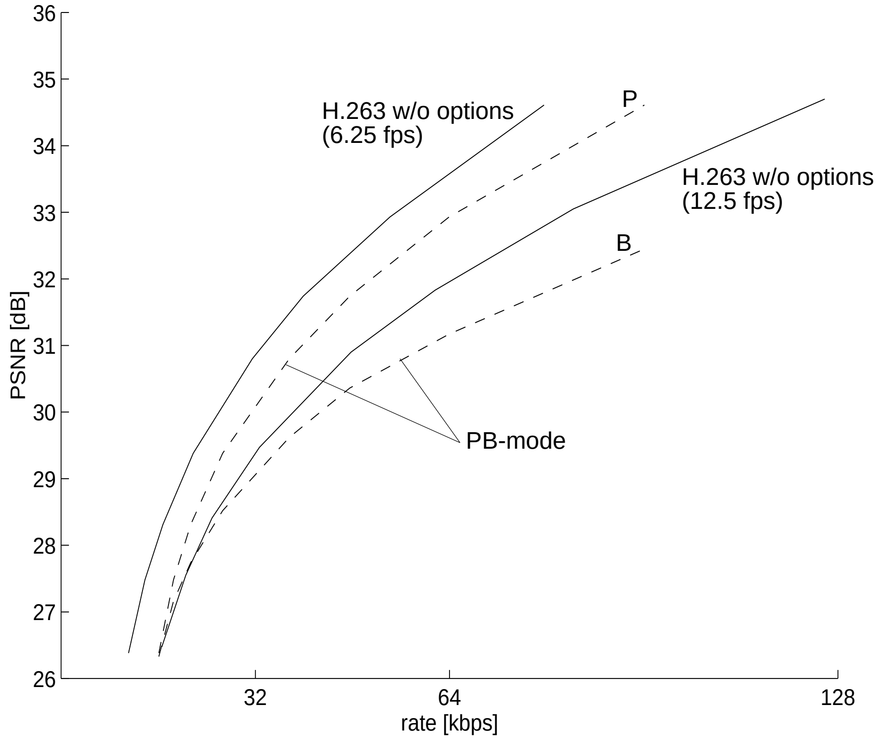

Figure 8: Performance of the H.263 option 'PB-frames mode' (Foreman).

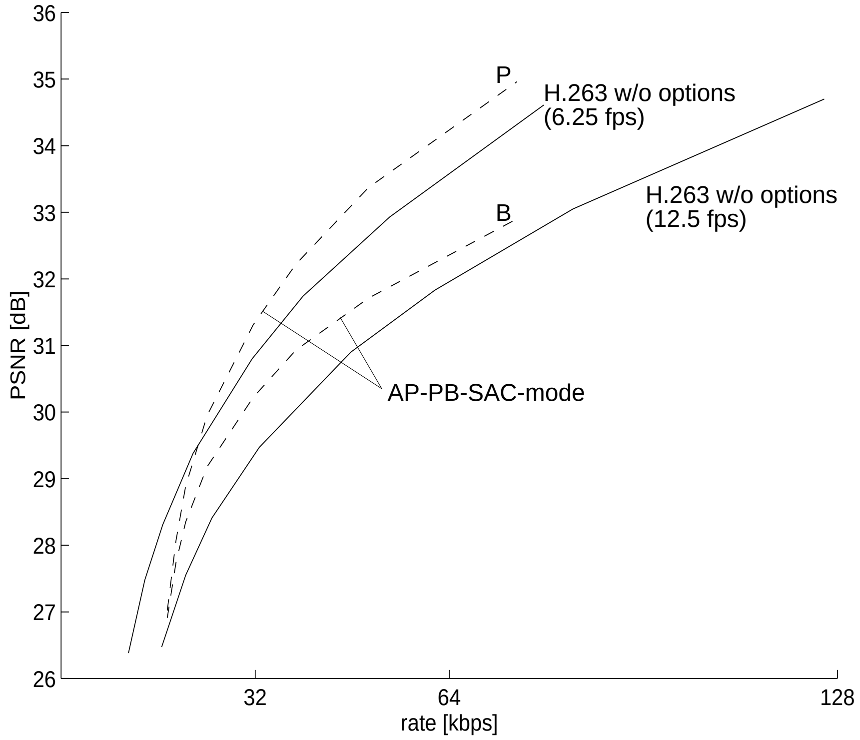

Figure 9: Performance of a combination of all H.263 options (Foreman).

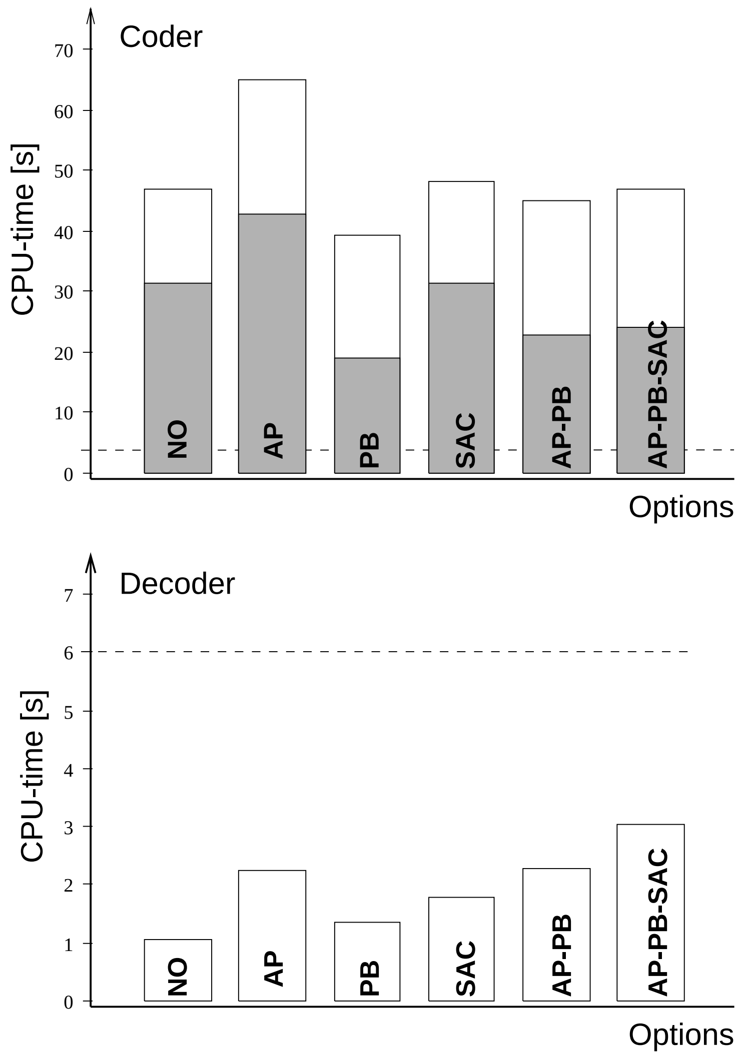

Figure 10: Comparison of the CPU-time consumed for H.263 with single options or combinations of them. The dashed line represents real-time. Gray bars indicate the amount of time consumed by motion estimation.

Bernd Girod received his M. S. degree in Electrical Engineering from Georgia Institute of Technology, in 1980 and his Doctoral degree "with highest honours" from University of Hannover, Germany, in 1987. Until 1987 he was a member of the research staff at the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover, working on moving image coding, human visual perception, and information theory. In 1988, he joined Massachussetts Institute of Technology, Cambridge, Mass., USA, first as a Visiting Scientist with the Research Laboratory of Electronics, then as an Assistant Professor of Media Technology at the Media Laboratory. From 1990 to 1993, he was Professor of Computer Graphics and Technical Director of the Academy of Media Arts in Cologne, Germany, jointly appointed with the Computer Science Section of Cologne University. He was a Visiting Adjunct Professor with the Digital Signal Processing Group at Georgia Institute of Technology, Atlanta, Ga., USA, in 1993. Since 1993, he is a Professor of Electrical Engineering/Telecommunications at University of Erlangen-Nuremberg, Germany, and the Head of the Telecommunications Institute. His research interests include multidimensional signal processing, information theory, video signal compression, human and machine vision, sensory computing, computer graphics and animation, as well as interactive media. For several years, he has served as a consultant to companies and government agencies. He is also co-founder and Chief Scientist of Vivo Software, Inc., Waltham, MA.

Eckehard Steinbach was born in Germany on August 27, 1969. He studied electrical engineering at University of Karlsruhe (Germany), University of Essex (Great-Britain) and ESIEE (Paris). After receiving his diploma in 1994 he joined the image communication group at the telecommunications institute of the University of Erlangen-Nuremberg (Germany), where he is currently working on his Ph.D. thesis. His research interests include image processing, computer vision and the extension of video coding standards for mobile applications.

Nikolaus Färber received his diploma in Electrical Engineering from the University of Munich, Germany, in 1993. Since 1994 he is a member of the research staff at the Telecommunications Institute of the University of Erlangen-Nuremberg, Germany. As a member of the image communication group he is currently working on his Ph.D. thesis on standard compatible video transmission in mobile environments and packet networks. Further research interests are combined source and channel coding and software only video coding.