A Software-based High-Quality MPEG-2 Encoder Employing Scene Change Detection and Adaptive Quantization

Dirk Farin[1] & Niels Mache[2] & Peter H.N. de With[3]
[1]Department of Computer Science IV, Univ. Mannheim, Germany,
[2]struktur AG, Germany, [3]CMG/Univ. Technol. Eindhoven, Netherlands

Abstract

This paper presents a software-only MPEG encoder implementation which uses adaptive quantization and scene change detection to enhance the image quality. Scene change detection is coupled to the bit-allocation process, providing a more constant image quality over time. Simultaneously, it is used to assign picture coding types for enabling easy, lossless cutting at scene change positions in a later editing process. Furthermore, a new fast bit-rate estimation algorithm is proposed, which is accurate enough to avoid macroblock-level rate-control. Although our current implementation concentrates on an MPEG-2 implementation, all concepts are readily applicable to MPEG-4 encoders.*

I. Introduction

In the field of MPEG encoding, real-time hardware encoders are widely used. However, for offline encoding applications like DVD authoring, software encoders offer the opportunity to use more memory and time on image content analysis to provide a better image quality.

An important aspect of video encoders is the quantization sub-system, comprising bit-allocation, rate-control, and adaptive quantization. Bit-allocation is usually implemented based on the techniques described in the so-called Test Model 5 (TM5) [17]. However, this algorithm is known to perform poorly at abrupt scene changes. Hence, various modifications have been published [18], [13] which introduce new I-frames after the scene change and adjust the bit-allocation appropriately. We propose a new modification which not only prevents quality degradation after scene changes, but even improves image quality by exploiting the temporal masking effect. Furthermore, it enables easy editing operations at scene change positions in the compressed domain.

Popular approaches for rate-control are based on feed-forward control [2], Lagrange multiplier based dynamic-programming [3], or estimates of the rate-distortion characteristic [5] to choose quantization-scale factors. Feed-forward control usually exposes difficulties with discontinuities in the input signal characteristics. In the case of the TM5 algorithm, this can even lead to unequal quality distribution in a single picture. Lagrange-multiplier based approaches find optimal solutions to the quantization problem, but are very computationally complex. Several approaches have been published which approximate the rate-distortion characteristic by a parametric model [6]. However, most techniques require that the model parameters have to be adapted, which usually means that the frame has to be encoded at several control points to determine the model parameters. Moreover, most models are frame-based and do not consider that an MPEG coded frame can consist of intra coded blocks as well as inter coded blocks. As the distribution of block modes may vary in a sequence of frames, the model accuracy is reduced.

We use a bit-rate estimation model that is based on macroblock units and that differentiates between macroblock coding modes. This enables an accurate estimation for coded frame size without requiring trial encodings to adjust model parameters. Our algorithm is an extension of the approach in [10]. Besides the number of non-zero DCT coefficients to estimate the bit-rate, we also use the value of the coefficient in our model.

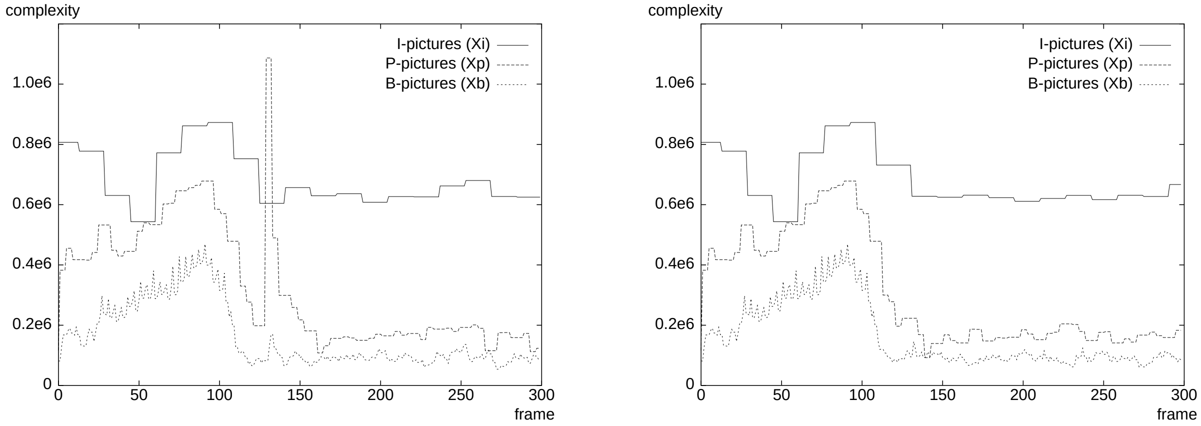
Adaptive quantization algorithms that are based on accurate models of the human visual system are too computationally complex for practical implementations. Hence, our adaptive quantization algorithm is based on low-level image features and is computationally efficient.

The paper is organized as follows: first, we give a short introduction into the TM5 bit-assignment algorithm on which our encoder is based. Subsequently, we modify this algorithm to take scene changes into account, and we introduce a new adaptive quantization algorithm which we are using instead of the TM5 technique. We describe our bit-rate estimation algorithm and show how scene adaptive bit-allocation, adaptive quantization, and rate-control can be combined into a complete system. Finally, we note some implementation issues and present results.

II. Bit-Allocation in TM5

Since our bit-allocation algorithm is based on the TM5 algorithm, we briefly describe the TM5 bit-allocation technique in this section. The TM5 bit-allocation is based on the assumption that the coded

*Part of this work has been carried out at the Image Understanding Department of the Institute of Parallel and Distributed High-Performance Systems at the University Stuttgart, Germany



(a) Estimated picture complexity without scene change detection

(b) Estimated picture complexity with enabled scene change detection

Fig. 1. Estimated picture complexities for the "table-tennis" sequence. This sequence contains a sudden scene change after frame 130. The coding parameters are a nominal GOP size of 16, P-distance 4 and a bit-rate of 850 kbit/s.

bit-rate can be estimated by a simple model as:

$$B_t = \frac{X_t}{Q_t} \qquad t \in \{i, p, b\},$$

where B_t is the number of bits required to code the picture, Q_t is the average quantization scale and X_t is a scene-dependent complexity parameter. The values X_i, X_p, and X_b are estimated by multiplying the average quantization scale and measured number of bits in the previously coded picture of the same coding type. This prediction is based on the assumption that the image content has slowly varying statistics. Hence, the bit-allocation process needs some time to stabilize after a large change of image content.

The intention of the TM5 bit-allocation process is to keep the ratio of quantization scales between different picture-coding types constant. The ratio is expressed by the constants $K_p = Q_p/Q_i$ and $K_b = Q_b/Q_i$. As each P-picture is used by several B-pictures as temporal prediction, an increase of P-picture quality also improves the quality of the dependent B-pictures. Consequently, bits spent to increase the quality of P-pictures result in a greater overall quality improvement as the same amount of bits spent on a B-picture. Hence, B-pictures are quantized slightly more coarsely and a $K_b > 1$ is used. The actual value of K_b is depending on the quantization matrices and B-picture distances. TM5 suggests the use of $K_p = 1$ and $K_b = 1.4$ for a P-picture distance of 3. Bits are assigned to each picture just before it is coded according to

$$B_t = \max\left\{ B'_t, \frac{bit\_rate}{8 \times picture\_rate} \right\} \qquad t \in \{i, p, b\},$$

$$B'_i(n) = \frac{R}{1 + N_p \frac{X_p}{X_i K_p} + N_b \frac{X_b}{X_i K_b}},$$

$$B'_p(n) = \frac{R}{N_p + N_b \frac{K_p X_b}{K_b X_p}}, \quad B'_b(n) = \frac{R}{N_b + N_p \frac{K_b X_p}{K_p X_b}},$$

where R is the number of bits remaining for coding the current GOP, and N_p, N_b are the number of uncoded P and B-pictures, respectively, remaining in the current GOP. The bit-allocation is lower bounded thereby ensuring that a minimum number of bits is assigned to each picture. Specifying a minimum bit-rate prevents extremely low-quality pictures in cases where the complexity estimation is much too low. This can for example happen when a complicated picture is coded after a black picture with very low complexity.

III. Scene Change Detection

The assumption that the scene statistics are varying slowly forms a fundamental limitation of the TM5 bit-allocation technique. This assumption is particularly not true when the video sequence contains sudden scene changes. In this section, we show how this problem can be alleviated using a simple scene change detection algorithm.

Consider the situation depicted in Figure 2 where an abrupt scene change occurs between the reference pictures P_3 and P_4. Because P_4 differs strongly from P_3, only few macroblocks in P_4 will find good temporal predictions in P_3. Consequently, a large number of macroblocks will be coded with intra-mode (fallback), thereby requiring more bits to code. Since the TM5

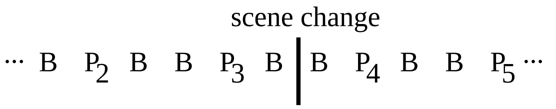


Fig. 2. Sudden scene change in the middle of a GOP.

bit-allocation scheme does not use look-ahead, it is not aware of the scene change and assigns the bits under the assumption that no change has occurred. This proposed bit-assignment will be too small to allow an adequate quality for P_4. Furthermore, the decreased quality of this P-picture will also decrease the quality of the surrounding predicted pictures because of the bad temporal prediction quality.

An additional problem arises in the calculation of the picture complexities. As the complexity for P_4 exceeds the usual P-picture complexity, X_p will be set to an unusually high value, which consequently will lead to an oversized bit-budget for the successive P-picture P_5.

This behaviour can be clearly observed in Figure 1a, which shows the estimated picture complexities as calculated by TM5. The sequence contains a sudden scene change at frame 130 and the increase in the following P-picture complexity is easily visible. There is also an increase in the estimated B-pictures complexities around the scene change, but this increase is not very significant because the B-pictures always have at least one appropriate reference frame for prediction.

A. GOP structure adjustment

To alleviate the previous difficulties, we propose to modify the structure of the group-of-pictures (GOP) at scene changes. After sudden scene changes, we enlarge or shrink the preceeding GOP to some extent so that the next GOP starts exactly with the first picture of the new scene (see Figure 3).

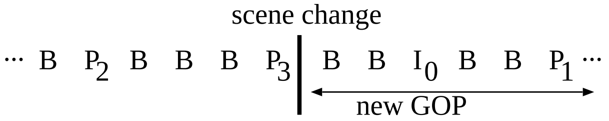


Fig. 3. Proposed GOP arrangement at a sudden scene change. The GOP after the scene change is coded as a closed-GOP. Note that the GOP preceding the scene change as been enlarged by one B-picture to align the end of the GOP with the scene change.

As the first picture of the new scene (in coding order) is now coded as an I-picture (as opposed to a P-picture in Figure 2), the estimation of P-picture complexity is not disturbed by the scene change and the bit-allocation stabilizes faster. Assuming that the I-picture complexity does not increase dramatically in the new scene, enough bits are assigned to the I-picture for enabling a good picture quality.

Figure 1b shows the estimated picture complexities of the same "table-tennis" sequence as in Figure 1a , but now with active use of our scene change detection. It can be seen that the mismatching peak in the P-picture complexity has disappeared and the estimation stabilizes very quickly for the new scene. This behaviour has a positive effect on the picture quality after the scene change. Figure 4 shows the measured PSNR for both disabled and enabled scene change detection under the same conditions. Obviously, without scene change detection, there is a sharp decrease of image quality after the scene change by about 5 dB. With enabled scene change detection, the decrease is both shorter in time and much smaller (about 1 dB). Figure 5 shows the difference in image quality of the picture following the scene change.

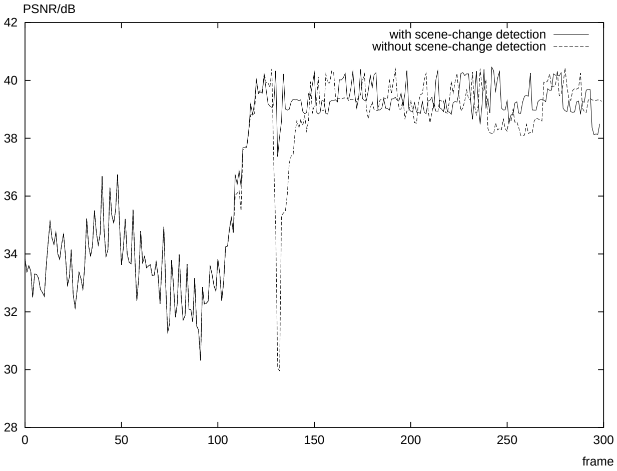


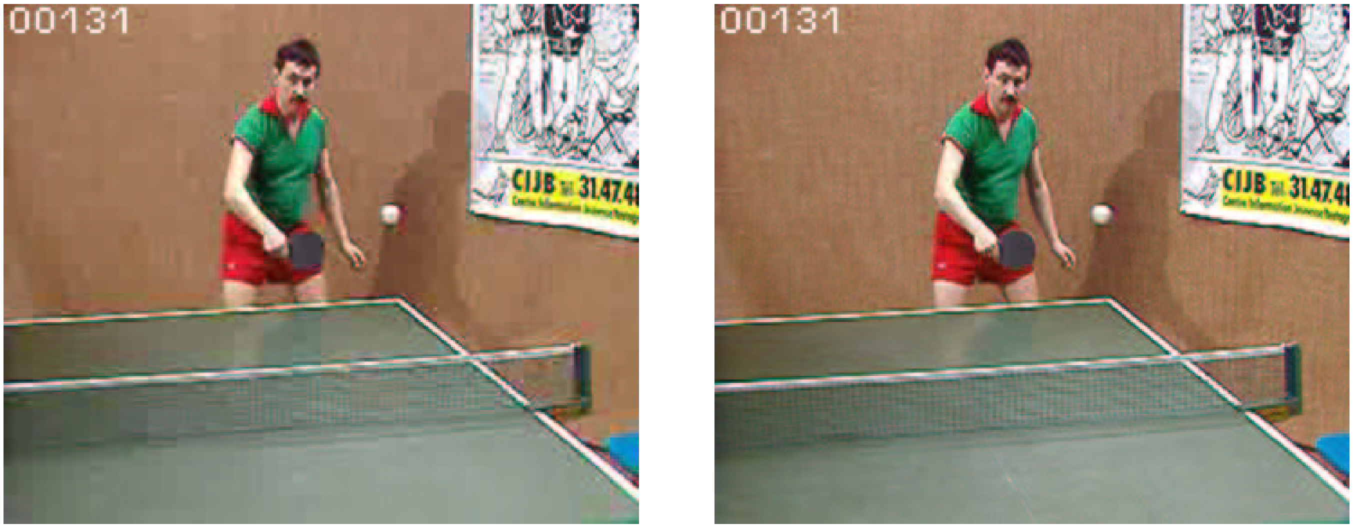
Fig. 4. Measured PSNR in dB for the "table-tennis" sequence, coded in CIF resolution at 850 kbit/s. A scene change occurs after frame 130.

B. Temporal Masking Effect

The new GOP structure also enables the utilization of the temporal masking effect of the human visual system. Temporal masking refers to the property of the human eye that it cannot perceive a picture at its full quality immediately after a sudden change. This effect can be exploited to code the first few pictures after a scene change with a lower quality without any perceptible degradation. When we use the GOP structure shown in Figure 3, this can be achieved by decreasing the number of bits assigned to the B-pictures between the scene change and the I-picture. The saved bits can now be used in the following pictures to increase their quality. We incorporate this temporal masking scheme into the TM5 framework by modifying the bit-allocation equations. For each (B-)picture with temporal reference n, we introduce an additional factor $K_{sc}(n)$ which indicates the strength of the temporal masking effect at picture n. Similarly to the factors K_p and K_b in the TM5 model, this factor modifies the ratio of average quantization scales for the pictures in a GOP. Larger values of $K_{sc}(n)$ lead to coarser quantization and consequently to a reduced image quality.

As $K_{sc}(n)$ varies for different B-pictures in a GOP, the bit-allocation equations of TM5 have to be generalized. Our modified bit-allocation equations take the new factor into account and are defined as

$$B_i'(n) = \frac{R}{1 + N_p \frac{X_p}{X_i K_p} + \sum_{i=n}^{s} \frac{X_b}{X_i K_b K_{sc}(i)}},$$



(a) No scene change detection

(b) With scene change detection

Fig. 5. Results of coding a picture immediately following a sudden scene change. Both pictures are coded as B-pictures under the same conditions. The picture at the right has a much better quality.

$$B'_p(n) = \frac{R}{N_p + \sum_{i=n}^{s} \frac{K_p X_b}{K_b K_{sc}(i) X_p}},$$

$$B'_b(n) = \frac{R}{N_p \frac{K_b K_{sc}(i) X_p}{K_p X_b} + \sum_{i=n}^{s} 1/K_{sc}(i)},$$

where s is the total GOP size. For simplicity of notation, we set $K_{sc}(n) = \infty$ for I and P-pictures. If $K_{sc}(n) = 1$ for all B-pictures, the formulas become exactly the allocation equations of TM5. Note that the modified bit-allocation now depends on the temporal reference of the picture since not all B-pictures are assigned the same amount of bits.

Because the temporal masking effect lasts for approximately 100ms, we increase $K_{sc}(n)$ to a value ≥ 1 for the B-pictures directly following the scene change up to the I-picture. For all other B-pictures, we set $K_{sc}(n) = 1$. Combined, we have:

$$K_{sc}(n) = \begin{cases} v \geq 1 & \text{for B-pictures after a scene ch.,} \\ 1 & \text{for all other B-pictures,} \\ \infty & \text{for I and P-pictures.} \end{cases}$$

The actual selection of a suitable v is subjective and has to be determined empirically. We have found that values of $K_{sc}(n) \approx 2$ give good results.

C. Simplified Edit Operations

Adapting the GOP structure to the scene changes also has an advantageous effect for video postprocessing. Note that the B-pictures after the scene change will not benefit from forward prediction, because the forward prediction reference-frame contains image content from the previous scene. Consequently, forward motion-prediction can be disabled for the B-pictures after a scene change and the GOP can be coded as a closed GOP. This allows the terminating GOP of the last scene and the first GOP of the new scene to be coded independently. A direct beneficial consequence is that splitting and concatenating the individual scenes in the MPEG stream is now possible without recompression. The practical consequence is that all edit operations at scene change positions can be performed in a computationally efficient way and without loss of quality.

D. Scene change detection

The objective of scene change detection in our application is to find favourable positions to start new GOPs. However, at the same time, we want to prevent the creation of very short or very long GOPs, which would decrease the overall coding performance. Hence, let s_{min} be the minimum allowed GOP-size and s_{max} the maximum GOP-size (typically, we set $s_{min} = 6$ and $s_{max} = 18$).

As sporadic detection errors only have a slight impact on image quality, a computationally efficient scene change detection algorithm can be used instead of more accurate, but computationally expensive algorithms. We chose to use a scene change detector based on brightness histogram differences [9]. This kind of detector works well with sudden changes, but it cannot detect slow transitions between scenes. However, for our application, this property is exactly desired, because we are only interested in sudden changes. Slow transitions cannot be coded more efficiently using a modified GOP structure and also do not trigger the temporal masking effect.

For the scene change detection, we use a look-ahead of $s_{max}+1$ pictures and compute the histogram differences $d(n, n+1)$ between all pairs of successive pictures. To find a scene change, we search for the maximum value of $d(n, n+1) \cdot w_n$ and classify it as a scene change if it is above a threshold t_u. The weighting factor w_n is used to favour GOP sizes near the nominal size m. It is defined as:

$$w_i = \begin{cases} \frac{1}{2}(1+(\frac{i-s_{min}}{m-s_{min}})^{\alpha}) & \text{for} \quad i \leq m \\ \frac{1}{2}(1+(\frac{s_{max}-i}{s_{max}-m})^{\alpha}) & \text{for} \quad i > m \end{cases}$$

where α can be adjusted to change the stringency of the encoder to use the nominal GOP size (see Fig. 6). If the maximum $d(n, n+1) \cdot w_n \leq t_u$, no scene change is detected and the nominal GOP size is used.

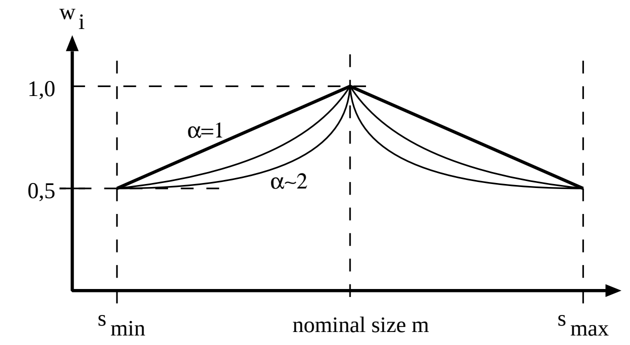


Fig. 6. Scene change detection weighting factor.

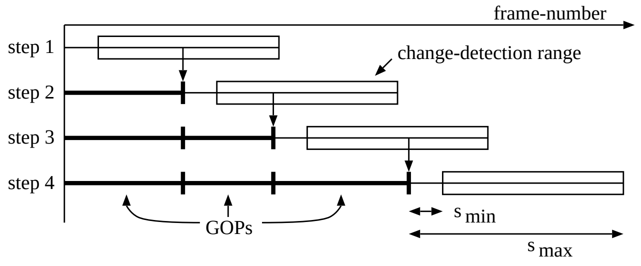


Fig. 7. GOP structure assignment process. The scene change detection range $[s_{min}; s_{max}]$ is searched for the maximum picture difference. The next GOP will be structured to end at this position. Subsequently, a new scene change detection is initiated beginning at this position.

E. Refinements

The robustness of our change detection algorithm can be further improved by introducing a second threshold t_l ($t_l < t_u$) to allow a third class for cases in which we cannot definitely decide if the observed difference has been caused by a scene change or by a fast moving scene. If the maximum histogram difference d_{max} is lower than t_l, we assume that no scene change is present and the nominal GOP size m is chosen for the GOP to be coded. This corresponds to the *no scene change* case defined above. If the maximum histogram difference lies in between the two thresholds, i.e. $t_l \leq d_{max} \leq t_u$, the GOP size is modified and optimized with respect to the scene change, but the temporal masking effect is ignored (i.e. $K_{sc}(n) \equiv 1$ for all B-pictures). This compromise is to benefit from the rearranged GOP structure in case that it is a scene change, and otherwise not to degrade the image quality in cases where no sudden scene change occurs.

IV. Adaptive Quantization

Adaptive quantization is applied at the macroblock level to reduce the amount of quantization noise in areas where it is most visible to the Human Visual System (HVS). The additional bits which are needed to provide the increased accuracy are obtained by reducing image quality in areas with fine, high-contrast texture (high-activity areas). The HVS is less sensitive to additional noise in these areas and cannot perceive the quality reduction.

A. Algorithm

Although our algorithm is not based on a discrete macroblock classification, let us consider the following three types of blocks:

- *Flat* blocks with only low detail. These blocks usually occur in uniform backgrounds like the sky in a natural scene. The HVS is sensitive to blocking artifacts in regions of flat blocks. Hence, a fine quantization scale should be chosen. As these regions only contain a few, low-frequency coefficients, the number required bits is small.
- *Textured* blocks containing fine detail texture with high variance. This corresponds to textured surfaces (e.g. grass), and regions with many small objects that are too small to be clearly visible. Because of the high frequencies in these blocks, they need a large amount of bits to be coded. Quantization noise in such areas is hardly visible and the quality can be decreased.
- *Mixed* blocks containing both flat and textured areas. These are usually boundary blocks between flat and textured areas. The texture areas in these blocks generate high-frequency coefficients which are quantized coarsely. Unfortunately, the quantization noise of those high-frequency coefficients can be easily perceived in the flat areas of the block, showing the typical ringing effect at object boundaries. Hence, these mixed blocks should be quantized with a fine step-size to reduce ringing artifacts.

The fundamental approach of our adaptive quantization is to move bits from image regions with textured blocks to mixed blocks, while the amount of bits assigned to flat blocks should not be changed. Since the transition between block types is smooth, we determine a real-valued *qnoise* indicator for each MB, which serves as a measure for predicting the perceptual visibility of

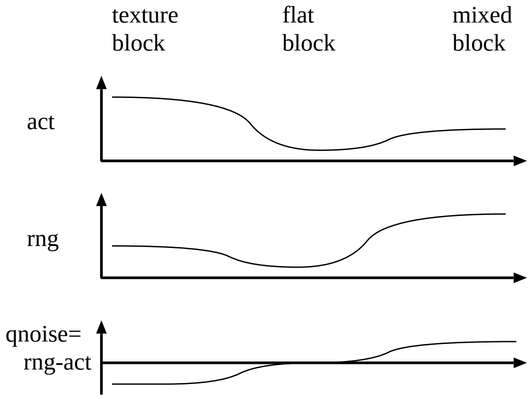


Fig. 8. Principle used for adaptive quantization. Textured blocks will receive a negative $qnoise$ as $act > rng$. Mixed blocks will receive a positive $qnoise$ because $act < rng$. In flat blocks act and bsy approximately cancel out.

the quantization noise. The $qnoise$ is evaluated by

$$qnoise = \alpha \cdot rng^{\gamma} - \beta \cdot act^{\delta},$$

where $\alpha, \beta, \gamma, \delta$ are empirically determined constants. The feature act is a measure of the *activity* in a block, which is high for high-frequency textures and low for flat blocks. The parameter rng is a feature describing the amount of ringing noise that is expected to be visible in the block. Figure 8 schematically depicts the relation of the three measures. Texture blocks will receive a low value of $qnoise$, mixed blocks will receive a high value, and flat blocks are assigned a value of about zero.

To determine appropriate values for the measures act and rng, we partition each macroblock into 4×4 *sub-blocks* of size 4×4 pixels each. For each sub-block located at (x, y), we calculate a sub-block activity as

$$subact_{(x,y)} = \sum_{\substack{0\le i\le 3\\ 0\le j\le 3}} \left|\frac{\partial f}{\partial x}(x+i, y+j)\right| + \left|\frac{\partial f}{\partial y}(x+i, y+j)\right|.$$

This sub-block activity is high for textured sub-blocks and low for flat sub-blocks. Subsequently, for each macroblock located at (x_0, y_0), we define

$$act_{(x_0,y_0)} = \sum_{\substack{0\le i\le 3\\ 0\le j\le 3}} subact_{(x_0+4i, y_0+4j)},$$

and

$$rng_{(x_0,y_0)} = \sum_{(x_a,y_a,x_b,y_b)\in P} |subact_{(x_a,y_a)} - subact_{(x_b,y_b)}|,$$

with $P = \Big\{(i,j,i,j+4), (j,i,j+4,i) \,|\, i \in \{0,4,8,12\},$ $j \in \{0,8\}\Big\}$ (see Fig. 9).

Adaptive quantization is applied by subtracting $qnoise$ from the quantization control parameter $MQUANT$ used in the MPEG coding process.

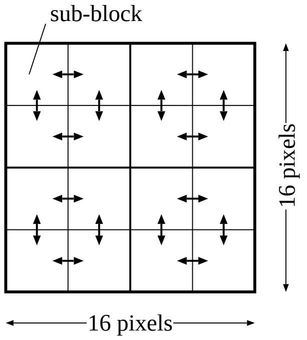


Fig. 9. Calculation of rng measure: absolute difference between each pair of sub-block activities indicated by double arrows is calculated and all differences are summed up.

B. Results

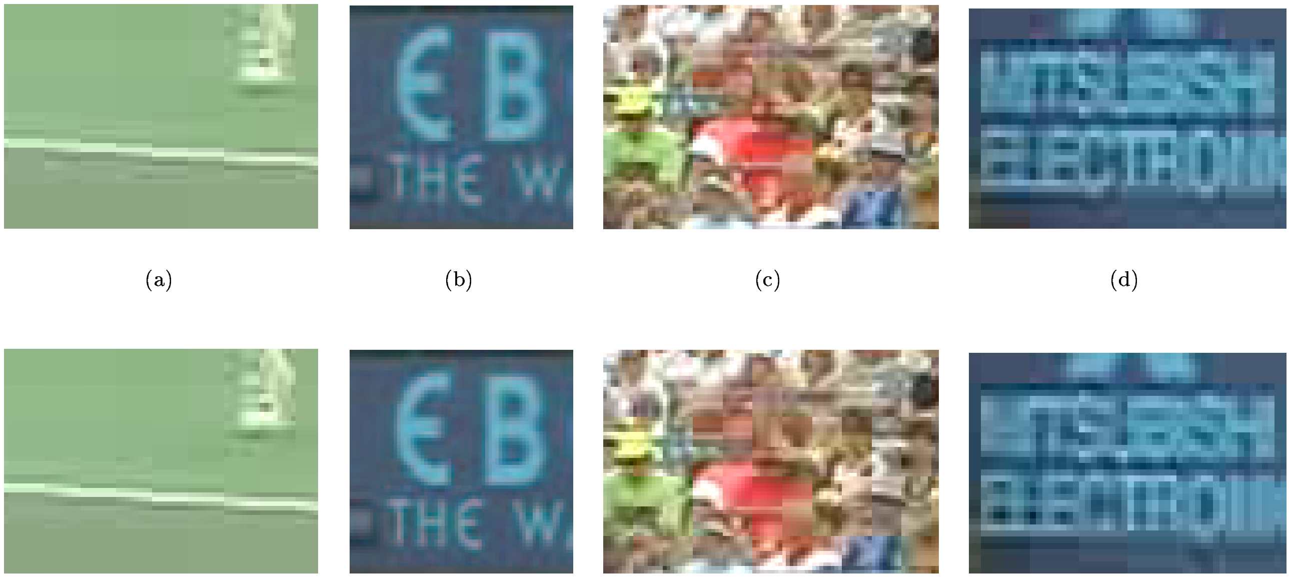
Figure 10 shows four regions of a frame from the "stefan" test-sequence, each coded with constant quantization and with adaptive quantization at the same bitrate. The quality of the regions 10(e),(f) is clearly improved by the adaptive quantization algorithm. The bits used to improve these regions were taken from the high activity area of the picture 10(g). However, the reduction of quality in this region is more difficult to perceive.

We observed a disadvantage of our algorithm in image areas with small text. The algorithm treats the text as high-activity area and reduces its quality (see Fig. 10(h)). This effect can be eliminated by integrating a text-detection algorithm which extracts areas containing text and prevents increasing the quantization scale in those areas.

V. Rate-Control

The task of the rate-control is to find a suitable quantization-scale so that the assigned number of bits is generated. One approach is to use the bit-rate model described in Section II. However, this model is not accurate enough to assure VBV compliance without recompression. The TM5 approach to achieve a sufficient accuracy is to observe the buffer fullness while coding the picture and to modify the quantization-scale to compensate the error. This macroblock-layer rate-control has two difficulties:

- The quantization control parameter is determined from the difference between the real buffer fullness and a virtual buffer fullness. To compute the virtual buffer fullness, a spatially uniform distribution of the bits in a picture is assumed. This assumption is not true for most natural images. Consider a scene with a sky that can be coded with only a few bits and a fine-textured ground requiring more bits. As TM5 tries to distribute the bits evenly across the image, the sky is coded with more bits than needed, while on the other hand, the ground will be assigned insufficient bits for an adequate quality.
- At low bit-rates, a frequent change of the quantizer



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 10. Illustration of adaptive quantization. Pictures (a)-(d) are based on constant quantization-scale, while pictures (e)-(h) result from using adaptive quantization.

step-size results in coding overhead and reduces overall image quality.

Because of these disadvantages, a better approach is to find a single, constant reference quantization step-size that will result in a picture size close to the allocation. Clearly, this is only achievable with a more accurate bit-rate model. Comparisons with optimal quantization algorithms [7] show that using frame-constant quantization scales comes close to the optimum achievable image quality.

A. Bit-Rate Estimation

Our bit-rate estimation model simplifies the problem of finding a bit-rate estimate for a whole picture by separately estimating the number of bits for each DCT-block. Apart from the fact that the estimation can be made more accurate for this small coding unit, this also allows an estimation when the quantization-scale is not constant in the picture and different coding-modes are used.

Since we consider the trial encodings at several quantization-scales for only adjusting the model parameters as a too costly task, we use features which can be directly and easily extracted from the available data. Experiments have shown that an accurate estimation can be calculated for each DCT coefficient block using only the unquantized coefficients as features.

For each intra-coded block (inter-coded blocks will be considered later), we calculate the absolute sum of all DCT coefficients $s(u,v)$ according to

$$S_I = \sum_{\substack{u,v \in [0;7], \\ (u,v) \neq (0,0)}} |s(u,v)|.$$

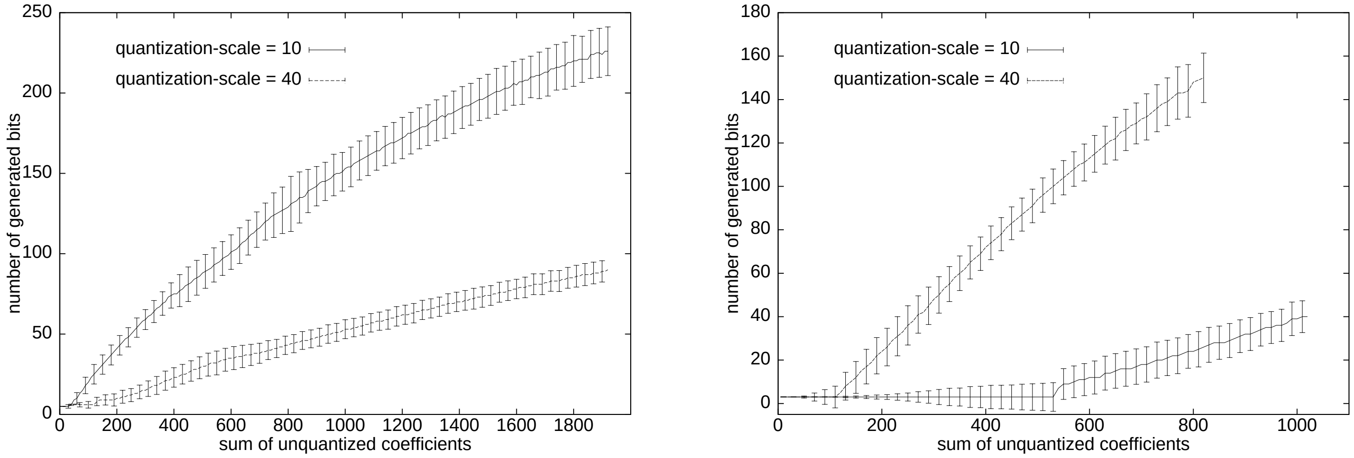
As the DC-coefficient in intra-coded blocks is coded independently using a DPCM coder, it is excluded from the estimation.

To determine the function $f_I(S_I, q) = b$ mapping the feature S_I to the number of bits b for a quantization-scale q, we coded several test-sequences using fixed quantization-scales and measured the number of bits generated for each DCT block. For each pair of S_I and q, f_I was set to the mean number of measured bits. To reduce the amount of data and fill undefined values, a piece-wise linear approximation of f_I was calculated for each q. Figure 11a shows $f_I(S_I, q)$ for $q = 10$ and $q = 40$, together with the standard deviation of the measured data.

Because the input-data statistics of inter-coded blocks differs from intra-coded blocks and different quantization matrices are used, we use a separate estimation for inter-coded blocks. The estimation is based on the same technique with the difference that for inter-blocks the DC-coefficient is also included in the sum of coefficients:

$$S_P = \sum_{u,v \in [0;7]} |s(u,v)|.$$

For each picture, the number of bits can now easily



(a) Intra-coded blocks

(b) Inter-coded blocks

Fig. 11. Measured number of bits per DCT-block for single DCT blocks at different quantization-scales. The standard deviation is portrayed by the small vertical bars.

be approximated by

$$bits(q) = \sum_{\text{DCT blocks}} \begin{cases} f_I(S_I, q) & \text{for intra coded blocks,} \\ f_P(S_P, q) & \text{for inter coded blocks.} \end{cases}$$

B. Offset-Compensation

Figure 13 shows the accuracy of the estimation on a sequence with several scene changes. Although the estimation accuracy is good, it can be seen that the estimated bit-rate differs from the actual bit-rate by a constant offset which seems to be scene dependent. This offset is caused by fixed header information, coding-mode flags, and motion-vectors (for inter-coded blocks), which are not included in the estimation. Furthermore, different input image statistics seem to have an influence on the estimation offset.

After a picture has been coded, the estimated number of bits as well as the actual coded image size is known and the offset can be determined. We compensate the offset by adding an estimate of the offset to the bit-rate estimation, which is obtained from previously coded pictures. The offset is applied at the macroblock level, with separate offsets for intra-mode macroblocks and inter-mode macroblocks. To adapt the offset to changing image content, it is calculated based on a moving average over a number of previously coded blocks of the same coding type (see Figure 12). The average has to be computed on enough blocks to cover several frames for preventing an oscillation of the offset estimation at scene cuts where the type of image content changes suddenly.

C. Results

We have evaluated our bit-rate estimation algorithm by coding the concatenation of the test sequences “table-tennis”, “salesman”, “deadline”, and “stefan” with constant quantization. In a first experiment, we coded the sequence with I-frames only. With enabled offset compensation, the maximum deviation of the estimation from the real size is only about 2% (see Fig. 13a). These maxima are reached when the image content changes rapidly because of scene changes or fast motion. When the image content is changing only slowly, the estimation error is neglegible.

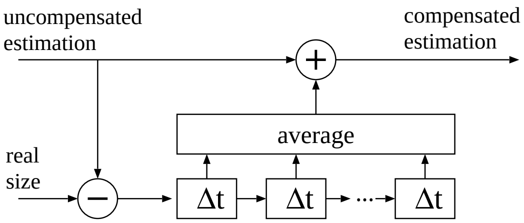
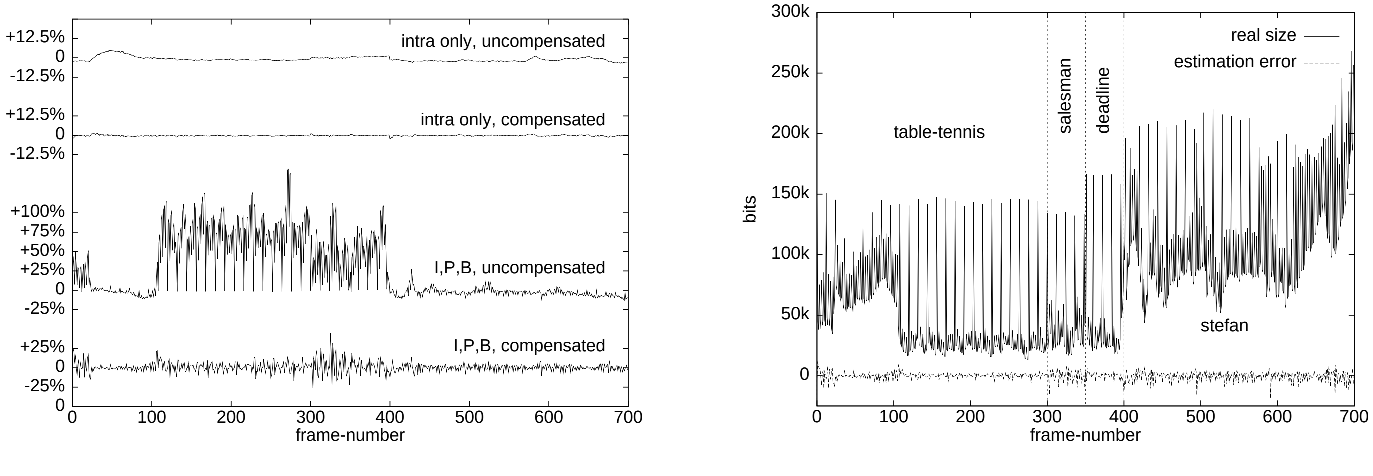


Fig. 12. Compensation of offset between bit-rate estimation and coded bit-rate.

In the second experiment, we coded the sequence with all frame-types. The estimation error of P and B-frames is considerably higher because many bits in predicted blocks are used for coding motion-vectors and only few are used for coding the residual. As our estimation only considers the bits used for coefficient coding, it is clear that the deviation is much larger. However, when activating offset compensation, the number of bits used for coding the motion-vectors get approximated by the estimation offset. Consequently, the offset compensation achieves to improve estimation accuracy such that the typical deviation is only about 10%.



(a) Relative estimation error

(b) Real-size and estimation error

Fig. 13. Estimation accuracy. The test-sequences “table-tennis”, “salesman”, “deadline”, and “stefan” have been concatenated to show the robustness at hard scene cuts.

VI. The total quantization control system

Figure 14 depicts how our rate-control and adaptive quantization is integrated into the encoder. The *block feature computation* unit pre-calculates the features S_I, S_P used for rate-control as well as rng, act used for adaptive quantization. The *adaptive quantization* block uses these pre-computed features to determine an $MQUANT$-modulation value. If images are coded that do not contain high-activity areas, adaptive quantization cannot operate favourably, since there are no areas where bits could be saved. The block *deactivate adaptive quantization* detects this case by summing up all act values in the image. If the sum is below a threshold, the $MQUANT$-modulation is switched off.

Rate-control is realized by adding a picture-level constant offset to the $MQUANT$-modulation such that the output image is coded with the amount of bits assigned in the bit-allocation stage. To determine the appropriate $MQUANT$ offset, an iterative process repeatedly estimates the number of bits and adjusts the offset until the allocation is reached. Bit-rate estimation is realized with the technique described in Section V. Note that this can be computed very efficiently based on the pre-computed features with simple table-lookups. The block *rate control* compares the number of generated bits with the allocation and modifies the $MQUANT$ offset accordingly until the actual size comes sufficiently close to the allocation.

VII. Implementation Aspects

Our software implementation is independent of a specific system type and has been tested on Intel x86, Sun Sparc, and ARM based systems. On Pentium-based processor architectures, the software makes use of MMX and MMX-2 SIMD instructions in time-critical parts like DCT, quantization, motion-estimation, and motion-compensation.

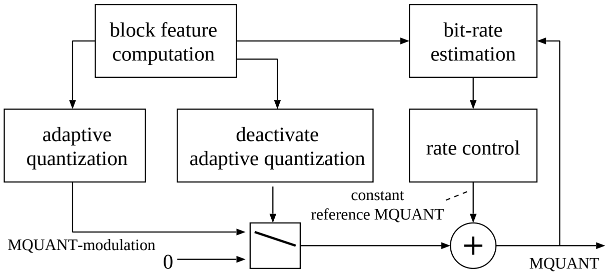

block feature computation
bit-rate estimation
adaptive quantization
deactivate adaptive quantization
rate control
constant reference MQUANT
MQUANT-modulation
0
+
MQUANT

Fig. 14. Rate-control including adaptive quantization.

For the scalar DCT implementation, we have adopted the algorithm described in [1] which is a row-column based algorithm with a minimum number of multiplications. In contrast, the MMX implementation is based on a row-column approach with different fast DCT algorithms for the row and column transforms [4]. Applying a single fast algorithm for both row and column transforms requires two matrix transpositions, which would induce inefficient memory operations. By using two independent algorithms, each can use a matrix decomposition that is optimized for the coefficient memory access-pattern. Consequently, a 2D-DCT can be carried out without matrix transposition.

The encoder achieves real-time encoding of CIF-resolution video sequences on a 500 MHz Pentium-III system using the Three-Step-Search algorithm for motion estimation. Moreover, the encoder supports multithreading for increased performance on SMP systems (see [8] for more information).

		TM5	SAMPEG /wo s.c.	SAMPEG /w s.c.
table-tennis	@768 kbps	34.85 dB	36.70 dB	36.77 dB
	@1125 kbps	36.85 dB	38.64 dB	38.72 dB
	@1500 kbps	38.45 dB	39.85 dB	39.84 dB
stefan	@768 kbps	28.29 dB	28.96 dB	29.00 dB
	@1125 kbps	30.34 dB	31.18 dB	31.21 dB
	@1500 kbps	31.97 dB	32.98 dB	32.99 dB
claire	@768 kbps	44.24 dB	45.31 dB	45.31 dB
	@1125 kbps	44.97 dB	46.69 dB	46.69 dB
	@1500 kbps	45.68 dB	47.41 dB	47.41 dB

TABLE I

Overall encoder performance (PSNR). All sequences were encoded at CIF resolution.

VIII. Results

We have evaluated the performance of our encoder by coding several sequences at different bit-rates (Table I). We coded all sequences both with enabled scene change detection and without, and using the TM5 reference encoder implementation. The GOP structure was fixed at $N = 12, M = 3$ (GOP structure may differ for enabled s.c. detection). To prevent any performance differences resulting from different motion-estimation algorithms, full-search with a search-range of ±16 pixels was used.

An PSNR increase of 1.0–2.0 dB compared with TM5 can be observed at all sequences. The increase is due to our rate-control technique, which generates an almost constant MQUANT value in each frame. This equalizes image quality across each frame and leads to only small overhead for quantizer change.

The results of our scene change adaptive encoding are designed to model human perception and cannot be measured with PSNR. However, we nevertheless observe a small increase of PSNR which is due to adapting the GOP structure to the scene content. Note that scene change detection does not influence our results with the "claire" sequence, since scene changes are absent in this sequence and the image content is too stable to trigger the GOP structure adjustment.

IX. Conclusions

We have described a new adaptive quantization and rate-control algorithm for MPEG-2 encoders. Scene change detection is used to stabilize bit-allocation and to exploit the temporal masking effect favourably for increasing the perceived video quality. Moreover, adapting the GOP pattern to the video content simplifies editing operations at scene changes in the compressed domain.

Our bit-rate estimation has low computational complexity, yet achieves high accuracy with only about 2% deviation for I-frames and about 10% deviation for P and B-frames. Since the rate-control operates on picture-level basis, the image quality does not vary across the picture. This increases overall image quality because there is no temporal flicker and no bits are wasted for switching quantization scale in the image for the sole purpose of rate control. Finally, adaptive quantization can be easily integrated into the rate-control to increase perceived image quality.

The total quantization control system improves the TM5 model of MPEG-2 with 1–2 dB in picture quality. The employed scene change detection technique gives a marginal increase of the average SNR but a noticeable increase of the perceptual quality.

References

[1] Y. Arai, T. Agui, and M. Nakajima. A fast DCT-SQ scheme for images. *Trans. of the IEICE*, E 71(11):1095, November 1988.

[2] Chih-Feng Chang and Jia-Shung Wang. A stable buffer control strategy for MPEG coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(6):920–924, December 1997.

[3] Jinho Choi and Daechul Park. A stable feedback control of the buffer state using the controlled lagrange multiplier method. *IEEE Transactions on Image Processing*, 3(5):546–557, September 1994.

[4] Intel Corporation. A fast precise implementation of 8 × 8 discrete cosine transform using the streaming SIMD extensions and MMX instructions; AP-922, 1999.

[5] Peter H. N. de With and Stephan J. J. Nijssen. A buffer regulation concept for MC-DCT systems tuning to constant quantization. 14^{th} *Symposium on Information Theory in the Benelux*, pages 176–183, May 1993.

[6] Wei Ding and Bede Liu. Rate control of MPEG video coding and recording by rate-quantization modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(1):12–20, 1996.

[7] Dirk Farin, Michael Käsemann, Peter H. N. de With, and Wolfgang Effelsberg. Rate-distortion optimal adaptive quantization and coefficient thresholding for MPEG coding. 23^{rd} *Symposium on Information Theory in the Benelux*, 2002.

[8] Dirk Farin, Niels Mache, and Peter H. N. de With. SAMPEG, a scene adaptive, parallel MPEG-2 software encoder. *SPIE Visual Communications and Image Processing*, January 2001.

[9] Ralph M. Ford, Craig Robson, Daniel Temple, and Michael Gerlach. Metrics for scene change detection in digital video sequences. *IEEE International Conference on Multimedia Computing and Systems*, 1997.

[10] Tae-yong Kim, Byeong-hee Roh, and Jae-kyoon Kim. An accurate bit-rate control for real-time MPEG video encoder. *Signal Processing: Image Communication*, 15:479–492, 2000.

[11] Myeong-jin Lee, Soon-kak Kwon, and Jae-kyoon Kim. A

scene adaptive bitrate control method in MPEG video coding. *SPIE VCIP*, pages 1406–1416, February 1997.

[12] Wilfried Osberger, Anthony J. Maeder, and Neil Bergmann. A perceptually based quantization technique for MPEG encoding. *SPIE Human Vision and Electronic Imaging III*, 3299, January 1998.

[13] Sanggyu Park, Youngsun Lee, and Hyunsik Chang. A new MPEG-2 rate control scheme using scene change detection. *ETRI Journal*, 18(2), July 1996.

[14] Mark R. Pickering and John F. Arnold. A perceptually efficient VBR rate control algorithm. *IEEE Transactions on Image Processing*, 3(5):527–531, September 1994.

[15] A. Puri and R. Aravind. Adaptive perceptual quantization for video compression. *SPIE Visual Communications and Image Processing*, 1605:297–300, 1992.

[16] Ishwar K. Sethi and Nilesh Patel. A statistical approach to scene change detection. *SPIE Storage and Retrieval for Image and Video Databases*, 2420, February 1995.

[17] MPEG-2, Test Model 5 (TM5). *Doc ISO/IEC JTC1/SC29/WG11/93-225b*. Test Model Editing Committee , April 1993.

[18] L. Wang. Rate control for MPEG video coding. *SPIE VCIP*, 2501, 1995.

[19] P. H. Westerink, R. Rajagopalan, and C. A. Gonzales. Two-pass MPEG-2 variable-bit-rate encoding. *IBM Journal of Research and Development*, 43(4):471–488, July 1999.

Dirk Farin graduated in computer science and electrical engineering from the University of Stuttgart, Germany. In 1999, he became research assistant at the Department of Circuitry and Simulation at the University of Mannheim. He joined the Department of Computer Science IV at the University of Mannheim in 2001. He received a best student paper award at the Symposium on Information Theory in the Benelux in 2001 and several awards in German national competitions in mathematics and computer science (Bundeswettbewerb Mathematik and Informatik). He developed several popular Open-Source projects including an MPEG decoder, two MPEG encoders, libraries with computer-vision algorithms, and other image-processing software. For the BMBF project “L3 — Lifelong Learning”, he wrote a video-database application with automatic video-summary generation. His research interests include video compression, image segmentation, and content analysis.

Niels Mache, born 1964 in Stuttgart, Germany, received the Diploma (MSc) in computer science and technical biology from the University of Stuttgart. From 1993 to 1999 he was research assistant in the German Human Genome Project at the Institute of Parallel and Distributed High-Performance Systems (IPVR). He has served as an R&D engineer at Sony Telecommunication Research and Development Europe. Niels was the Director of R&D and founder of delix GmbH, a Linux operating system developer, distributor and ISP, in Stuttgart, Germany. The company’s Linux operations were acquired by Red Hat Inc. in June 1999. He was the Director of Development for Red Hat, Germany leading the development of Red Hat Linux 6.1 Professional Package, which was released in January 2000. Niels is the CEO and a founder of struktur AG, a software company focused on XML technology and content management. Niels has more than 50 international publications. He got awards from Jugend Forscht, German Academy Software and MasPar Computer. In addition, Niels holds several technology patents.

Peter H.N. de With graduated in electrical engineering from the University of Technology in Eindhoven. In 1992, he received his Ph.D. degree from the University of Technology Delft, The Netherlands, for his work on video bitrate reduction for recording applications. He joined Philips Research Labs Eindhoven in 1984, where he became a member of the Magnetic Recording Systems Department. From 1985 to 1993 he was involved in several European projects on SDTV and HDTV recording. In this period he contributed as a coding expert to the DV standardization. In 1994 he became a member of the TV Systems group, where he was leading the design of advanced programmable video architectures. In 1996, he became senior TV systems architect and in 1997, he was appointed as full professor at the University of Mannheim, Germany, at the faculty Computer Engineering. In 2000, he joined CMG Eindhoven as a principal consultant and he became professor at the University of Technology Eindhoven, at the Faculty Electrical Engineering and the embedded systems institute (EESI). He has written numerous papers on video coding, architectures and their realization. Regularly, he is a teacher of the Philips Technical Training Centre and for other post-academic courses. In 1995 and 2000, he co-authored papers that received the IEEE CES Transactions Paper Award. In 1996, he obtained a company Invention Award. In 1997, Philips received the ITVA Award for its contributions to the DV standard. Mr. de With is a senior member of the IEEE, program committee member of the IEEE CES, chairman of the Benelux Working Group on Information Theory, member of the scientific board of CMG, ASCI, and various other working groups.