

Affine Multipicture Motion-Compensated Prediction

Thomas Wiegand, *Member, IEEE*, Eckehard Steinbach, and Bernd Girod, *Fellow, IEEE*

Abstract—Affine motion compensation is combined with long-term memory motion-compensated prediction. The idea is to determine several affine motion parameter sets on subareas of the image. Then, for each affine motion parameter set, a complete reference picture is warped and inserted into the multipicture buffer. Given the multipicture buffer of decoded pictures and affine warped versions thereof, block-based translational motion-compensated prediction and Lagrangian coder control are utilized. The affine motion parameters are transmitted as side information requiring additional bit rate. Hence, the utility of each reference picture and, with that, each affine motion parameter set is tested for its rate-distortion efficiency. The combination of affine and long-term memory motion-compensated prediction provides a highly efficient video compression scheme in terms of rate-distortion performance. The two incorporated multipicture concepts complement each other well providing almost additive rate-distortion gains. When warping the prior decoded picture, average bit-rate savings of 15% against TMN-10, the test model of ITU-T Recommendation H.263, are reported for the case that 20 warped reference pictures are used. When employing 20 warped reference pictures and 10 decoded reference pictures, average bit-rate savings of 24% can be obtained for a set of eight test sequences. These bit-rate savings correspond to gains in PSNR between 0.8–3 dB. For some cases, the combination of affine and long-term memory motion-compensated prediction provides more than additive gains.

Index Terms—Affine motion model, H.263, multipicture motion compensation, video coding.

I. INTRODUCTION

THE most successful class of today's video compression schemes are called hybrid codecs. The concept of block-based motion-compensated prediction (MCP) is prevalent in all these coding schemes [1]. The achievable MCP performance can be increased by reducing the size of the motion-compensated blocks [2]. However, the bit rate must be assigned carefully to the motion vectors of these smaller blocks. Therefore, rate-constrained motion estimation is often employed yielding improved compression efficiency [2], [3], [1]. In rate-constrained motion estimation, a Lagrangian cost function $J = D + \lambda R$ is minimized, where distortion D is weighted against rate R using a Lagrange multiplier λ . Moreover, the macroblock mode decision should also be based on Lagrangian optimization techniques [4].

Manuscript received November 3, 2000; revised July 20, 2002. This paper was recommended by Associate Editor H. Watanabe.

T. Wiegand is with the Image Processing Department, Fraunhofer Institute for Telecommunications—Heinrich-Hertz-Institut (HHI), 10587 Berlin, Germany (e-mail: wiegand@hhi.de).

E. Steinbach is with the Institute of Communication Networks, Media Technology Group, Munich University of Technology, 80290 Munich, Germany (e-mail: Eckehard.Steinbach@tum.de).

B. Girod is with the Information Systems Laboratory, Stanford University, Stanford, CA 94305 USA (e-mail: bgirod@stanford.edu).

Digital Object Identifier 10.1109/TCSVT.2004.841690

Long-term MCP [5], [6] increases the efficiency of video compression schemes by utilizing several past pictures that are assembled in a multipicture buffer. This buffer is simultaneously maintained at encoder and decoder. Block-based MCP is performed using motion vectors that consist of a spatial displacement and a picture reference to address a block in the multipicture buffer. Rate-constrained motion estimation is employed to control the bit rate of the motion data. The ITU-T Video Coding Experts Group (ITU-T/SG16/Q.6) has decided to adopt this feature as an Annex to the H.263 standard [7]. Moreover, the recent H.264/AVC video coding standard contains long-term MCP as a mandatory feature in all profiles [8], [9].

While long-term memory MCP extends the motion model to exploit long-term dependencies in the video sequence, the motion model remains translational. However, independently moving objects in combination with camera motion and focal length change lead to a sophisticated motion vector field which may not be efficiently approximated by a translational motion model. With an increasing time interval between video pictures, as is the case when employing long-term memory MCP, this effect is further enhanced since more sophisticated motion is likely to occur. Hence, the efficiency of coding the motion information is often increased by enhancing the motion model.

In an early work, Tsai and Huang derive a parametric motion model that relates the motion of planar objects in the scene to the observable motion field in the image plane for a perspective projection model [10]. The eight parameters of this model are estimated using corresponding feature points [10]. However, noisy feature point correspondences typically have a strong effect on the accuracy of the parameter estimate. In [11], Hoetter and Thoma approximate the planar object motion using a two-dimensional quadratic model of twelve parameters. The parameters are estimated using spatial and temporal intensity gradients which drastically improves the parameter estimates in the presence of noise.

Various researchers have utilized affine and bilinear motion models for *object-based* or *region-based* coding of image sequences, e.g., see [12]–[17]. The motion parameters are estimated such that they lead to an efficient representation of the motion field inside the corresponding image partition. Due to the mutual dependency of motion estimation and image partition a combined estimation must be utilized. This results in a sophisticated optimization task which usually is computationally demanding.

Other researchers have used affine or bilinear motion models in conjunction with a *block-based* approach to reduce the bit rate for transmitting the image segmentation [18], [19]. They have faced the problem that, especially at low bit rates, the overhead associated with higher order motion models that are assigned to smaller size blocks might be prohibitive. A combination of the

block-based and the region-based approach is presented in [20]. Karczewicz *et al.* report in [20] that the use of the twelve-parameter motion model in conjunction with a coarse segmentation of the video picture into regions, that consist of a set of connected blocks of size 8×8 samples, can be beneficial in terms of coding efficiency.

Within the MPEG-4 standardization group, a technique called *Sprites* has been considered [21]–[23]. Sprites can exploit long-term statistical dependencies similar to background memory techniques [24]–[26], [21], [27]. The advantage of Sprites is that they can robustly handle camera motion. In addition, image content that temporarily leaves the field of view can be more efficiently represented. The motion model used is typically a six-parameter affine model or an eight-parameter perspective model. The generation of the background mosaic is conducted either online or offline and the two approaches are referred to as *Dynamic Sprites* and *Static Sprites*, respectively. So far, only Static Sprites are part of the MPEG-4 standard [28]. For Static Sprites, an iterative procedure is applied to analyze the motion in a video sequences of several seconds to arrive at robust segmentation results. This introduces a delay problem that cannot be resolved in interactive applications. On the other hand, the online estimation problem for Dynamic Sprites has been very difficult to solve and with advantages being reported in [23].

An interesting generalization of the background memory and Sprite techniques has been proposed by Wang and Adelson, wherein the image sequence is represented by *layers* [29]. In addition to the background, the so-called *layered coding* technique can represent other objects in the scene as well. As for Static Sprites, the layers are determined by an iterative analysis of the motion in a complete image sequence of several seconds.

A simplification of the clustering problem in object-based or region-based coding and the parameter estimation in Sprite and layered coding is achieved by restricting the motion compensation to one global model that compensates for the camera motion and focal length changes [30]–[32]. Often, the background in the scene is assumed to be static and motion of the background in the image plane is considered to be camera motion. For the *global motion compensation* of the background, often an affine motion model is used where the parameters are estimated typically using two steps. In the first step, the motion parameters are estimated for the entire image, and, in the second step, the largest motion cluster is extracted. The globally motion-compensated picture is either provided additionally as a second reference picture or the prior decoded picture is replaced. Given the globally motion-compensated image as a reference picture, typically a block-based hybrid video coder conducts translational motion compensation. The drawback of global motion compensation is the limitation in rate-distortion performance due to the restriction to one motion parameter vector per picture. The benefits of this approach are the avoidance of sophisticated segmentation and parameter estimation problems. Global motion compensation is therefore standardized as an Annex of H.263 [7] and part of MPEG-4's Advanced Simple Profile [28] to enhance coding efficiency mainly for on-line encoding of video.

In this paper, the global motion compensation idea is extended to employing several affine motion parameter sets. The

estimation of the various affine motion parameter sets is conducted so as to handle multiple independently moving objects in combination with camera motion and focal length change. Long-term statistical dependencies are exploited as well by incorporating long-term memory MCP. The paper is organized as follows. In Section II, the extension of long-term memory MCP to affine motion compensation is explained. The coder control is described in Section III, where the estimation procedure for the affine motion parameters and the reference picture warping are presented. Then, the determination of an efficient number of affine motion parameter sets is described. Finally, in Section IV, experimental results are presented when incorporating affine multipicture MCP into an H.263 codec. The rate-distortion performance in comparison to TMN-10 (the test model of H.263) and the extension of TMN-10 by long-term memory MCP is provided.

II. AFFINE MULTIPICTURE MOTION COMPENSATION

In this section, affine multipicture motion compensation is explained. First, the extension of the multipicture buffer by warped versions of decoded pictures is described. Then, the necessary syntax extensions are outlined and the affine motion model, i.e., the equations that relate the affine motion parameters to the sample-wise motion vector field, are presented.

The block diagram of the multipicture affine motion-compensated predictor is depicted in Fig. 1. The motion-compensated predictor utilizes $M = K + N$ ($M \geq 1$) picture memories. The M picture memories are composed of two sets:

Set 1: K past decoded pictures.

Set 2: N warped versions of past decoded pictures.

The H.263-based multipicture predictor conducts block-based MCP using all $M = K + N$ pictures and produces a motion-compensated picture. This motion-compensated picture is then used in a standard hybrid DCT video coder [7], [1]. The N warped reference pictures are determined using the following two steps:

Step 1) Estimation of N affine motion parameter sets between the K previous pictures and the current picture.

Step 2) Affine warping of N reference pictures.

The number of efficient reference pictures $K \leq M^* \leq M$ is determined by evaluating their rate-distortion efficiency for each reference picture. The M^* chosen reference pictures with the associated affine motion parameter sets are transmitted in the header of each picture. The order of their transmission provides an index that is used to specify a particular reference picture on a block basis. The decoder maintains only the K decoded reference pictures and does not have to warp N complete pictures for motion compensation. Rather, for each block or macroblock that is compensated using affine motion compensation, the translational motion vector and the affine motion parameter set are combined to obtain the displacement field for that image segment.

Figs. 2 and 3 show an example of affine multipicture warping. The left-hand side of Fig. 2 is the most recent decoded picture that would be the only picture to predict the right-hand side of Fig. 2 in single-picture motion compensation. In Fig. 3,

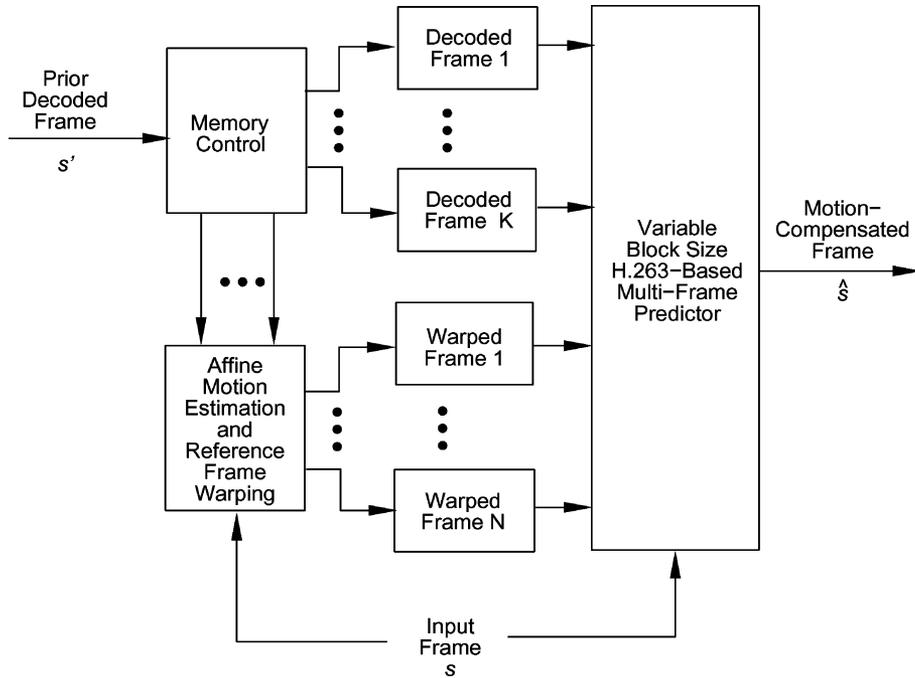


Fig. 1. Block diagram of the affine multipicture motion-compensated predictor.



Fig. 2. Two pictures from the QCIF test sequence *Foreman*: (a) previous decoded picture and (b) current picture to be encoded.

four out of the set of additionally employed reference pictures are shown. Instead of just searching over the previous decoded picture [Fig. 2(a)], the block-based motion estimator can also search positions in the additional reference pictures like the ones depicted in Fig. 3 and transmits the corresponding spatial displacement vectors and picture reference parameters.

A. Syntax of the Video Codec

Affine multipicture MCP is integrated into a video codec that is based on ITU-T Recommendation H.263 [7]. H.263 uses the typical basic structure that has been predominant in all video coding standards since the development of H.261 [33] in 1990, where the image is partitioned into macroblocks of 16×16 luma samples and 8×8 chroma samples. Each macroblock can either be coded in INTRA or one of several predictive coding modes. The predictive coding modes can either be of the types SKIP, INTER, or INTER+4V. For the SKIP mode, just one bit is spent to signal that the samples of the macroblock are repeated from the prior decoded picture. The INTER coding mode uses blocks of size 16×16 luma samples and the INTER+4V coding mode uses blocks of size 8×8 luma



Fig. 3. Four additional reference pictures. The upper left picture is a decoded picture that was transmitted two picture intervals before the previous decoded picture. The upper right picture is a warped version of the decoded picture that was transmitted one picture interval before the previous picture. The lower two pictures are warped versions of the previous decoded picture.

samples for motion compensation. For both modes, the MCP residual image is encoded similarly to INTRA coding by using a DCT for 8×8 blocks followed by scalar quantization of transform coefficients and run-level variable-length entropy coding. The motion compensation can be conducted using half-sample accurate motion vectors where the intermediate positions are obtained via bilinear interpolation.

In a well-designed video codec, the most efficient concepts should be combined in such a way that their utility can be adapted to the source signal without significant bit-rate overhead. Hence, the proposed video codec enables the utilization of variable block-size coding, long-term memory prediction and

affine motion compensation using such an adaptive method, where the use of the multiple reference pictures and affine motion parameter sets can be signaled with very little overhead. The parameters for the chosen reference pictures are transmitted in the header of each picture. First, their actual number M^* is signaled using a variable length code. Then, for each of the M^* reference pictures, an index identifying one of the past K decoded pictures is transmitted. This index is followed by a bit signaling whether the indicated decoded picture is warped or not. If that bit indicates a warped picture, the corresponding six affine motion parameters are transmitted. This syntax allows the adaptation of the multipicture affine coder to the source signal on a picture-by-picture basis without incurring much overhead. Hence, if affine motion compensation is not efficient, only one bit per reference picture header is needed to turn it off.

B. Affine Motion Model

In this study, an affine motion model is employed that describes the relationship between the motion of planar objects and the observable motion field in the image plane via a parametric expression. This model can describe motion such as translation, rotation, and zoom using six parameters $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)^T$. For estimation and transmission of the affine motion parameter sets, the orthogonalization approach in [20] is adopted. The orthogonalized affine model is used to code the displacement field $(\mathbf{m}_x[\mathbf{a}, x, y], \mathbf{m}_y[\mathbf{a}, x, y])^T$ and to transmit the affine motion parameters using uniform scalar quantization and variable length codes. In [20], a comparison was made to other approaches indicating the efficiency of the orthogonalized motion model. The motion model used for the investigations in this paper is given as

$$\begin{aligned} \mathbf{m}_x[\mathbf{a}, x, y] &= \frac{w-1}{2} \left[a_1 \cdot c_1 + a_2 \cdot c_2 \cdot \left(x - \frac{w-1}{2} \right) + a_3 \cdot c_3 \cdot \left(y - \frac{h-1}{2} \right) \right] \\ \mathbf{m}_y[\mathbf{a}, x, y] &= \frac{h-1}{2} \left[a_4 \cdot c_1 + a_5 \cdot c_2 \cdot \left(x - \frac{w-1}{2} \right) + a_6 \cdot c_3 \cdot \left(y - \frac{h-1}{2} \right) \right] \end{aligned} \quad (1)$$

in which x and y are discrete sample locations in the image with $0 \leq x < w$ and $0 \leq y < h$ and w as well as h being the image width and height, respectively. The coefficients c_1 , c_2 , and c_3 in (1) are given as

$$\begin{aligned} c_1 &= \frac{1}{\sqrt{w \cdot h}} \\ c_2 &= \sqrt{\frac{12}{w \cdot h \cdot (w-1) \cdot (w+1)}} \\ c_3 &= \sqrt{\frac{12}{w \cdot h \cdot (h-1) \cdot (h+1)}}. \end{aligned} \quad (2)$$

The affine motion parameters a_i are quantized as follows:

$$\begin{aligned} \tilde{a}_i &= \frac{Q(\Delta \cdot a_i)}{\Delta} \\ \Delta &= 2 \end{aligned} \quad (3)$$

where $Q(\cdot)$ means rounding to the nearest integer value. The quantization levels of the affine motion parameters $q_i = \Delta \cdot \tilde{a}_i$ are entropy-coded and transmitted. It has been found experimentally that similar coding results are obtained when varying the coarseness of the motion coefficient quantizer Δ in (3) from 2 to 10. Values of Δ outside this range, i.e., larger than 10 or smaller than 2, adversely affect coding performance. Typically, an affine motion parameter set requires between 8–40 b for transmission.

III. RATE-CONSTRAINED CODER CONTROL

In the previous section, the architecture and syntax for the affine multipicture video codec are described. Ideally, the coder control should determine the coding parameters so as to achieve a rate-distortion efficient representation of the video signal. Typical video sequences contain widely varying content and motion and can be more efficiently compressed if several different techniques are permitted to code different regions. For the affine motion coder, one additionally faces the problem that the number of reference pictures has to be determined since each warped reference picture is associated with an overhead bit rate. Therefore, the affine motion parameter sets must be assigned to large image segments to keep their number small. In most cases however, these large image segments usually cannot be chosen so as to partition the image uniformly. The proposed solution to this problem is as follows.

- Step A) Estimate N affine motion parameter sets between the current and the K previous decoded pictures.
- Step B) Generate the multipicture buffer which is composed of K past decoded pictures and N warped pictures that correspond to the N affine motion parameter sets, which are determined in **Step A**.
- Step C) Conduct multipicture block-based hybrid video encoding on the $M = N + K$ reference pictures.
- Step D) Determine the number of affine motion parameter sets that are efficient in terms of rate-distortion performance.

In the following, steps A)–D) are described in detail.

A. Step A: Affine Motion Parameter Estimation

A natural camera-view scene may contain multiple independently moving objects in combination with camera motion and focal length change often resulting in spatially inconsistent motion vector fields. Hence, region-based coding attempts to segment the motion vector field into consistent regions. In this study, such an explicit segmentation of the scene is avoided. Instead, the image is partitioned into N blocks of fixed size which are referred to as *initial clusters* in the following. For each initial cluster one affine motion parameter set is estimated that describes the motion inside this block between a decoded picture and the current original picture. The estimation of the affine motion parameter set for each initial cluster is conducted as follows.

- 1) **Initialization of the affine refinement:** estimation of L initial translational motion vectors, in order to robustly deal with large displacements.

- 2) **Affine refinement:** for each of the L initial translational motion vectors, computation of an affine motion parameter set using an image intensity gradient-based approach. The affine motion parameters are estimated by solving an over-determined set of linear equations so as to minimize MSE.

Finally, the best affine motion parameter set in terms of MSE is chosen among the L considered candidates. In the following, the steps for affine motion estimation are discussed in detail.

1) *Initialization of the Affine Refinement:* For the initialization of the affine refinement step, two methods are discussed:

- *cluster-based initialization;*
- *macroblock-based initialization.*

For the *cluster-based initialization*, the picture is partitioned into N blocks of identical size. In our implementation, we have chosen $N = 1, 2, 4, 8, 16, 32, 64,$ and 99 clusters corresponding to blocks of size $176 \times 144/N$ samples for QCIF pictures. The purpose of this initialization method is the simple experimental determination of an efficient number of initial clusters, since this approach provides that flexibility. Hence, it will be used in Section IV to analyze the tradeoff between rate-distortion performance and complexity that is proportional to the number of initial clusters N since this number is proportional to the number of warped reference pictures. For this initialization method, the MSE for block matching is computed over all samples inside the cluster to obtain one initial motion vector for each reference picture. Hence, the number L of initial translational motion vectors per cluster that are considered for the refinement step is identical to the number of decoded reference pictures K . This imposes a computational burden that increases as the number of decoded reference pictures K grows since the affine refinement routine is repeated for each initial translational motion vector.

Please note that, in H.263 and the long-term memory MCP coder, translational motion estimation has to be conducted anyway for 16×16 blocks. Hence, reusing those motion vectors for initialization of the affine refinement would help to avoid the extra block matching step of the cluster-based initialization. This approach is called the *macroblock-based initialization*. For that, an image partitioning is considered where the clusters are aligned with the macroblock boundaries. An example for such an initial partitioning is depicted in Fig. 4. Fig. 4 shows a QCIF picture from the sequence *Foreman* that is superimposed with 99 blocks of size 16×16 samples. The $N = 20$ clusters are either blocks of size 32×32 samples comprising four macroblocks, or blocks of size $32 \times 48, 48 \times 32,$ or 48×48 samples. For macroblock-based initialization, the motion vector of each macroblock is utilized as an initialization to the affine refinement step, and therefore either $L = 4, 6$ or 9 initial translational motion vectors are considered. This number is independent from the number of decoded reference pictures K . To obtain an initial motion vector $\mathbf{m}^I = (\mathbf{m}_x^I, \mathbf{m}_y^I, \mathbf{m}_t^I)^T$ which contains the spatial displacements \mathbf{m}_x^I and \mathbf{m}_y^I as well as the picture reference parameter \mathbf{m}_t^I , a Lagrangian cost function is minimized which is given as

$$\mathbf{m}^I = \arg \min_{\mathbf{m} \in \mathcal{M}} \{D_{DFD}(\mathbf{S}_k, \mathbf{m}) + \lambda \cdot R(\mathbf{S}_k, \mathbf{m})\}. \quad (4)$$

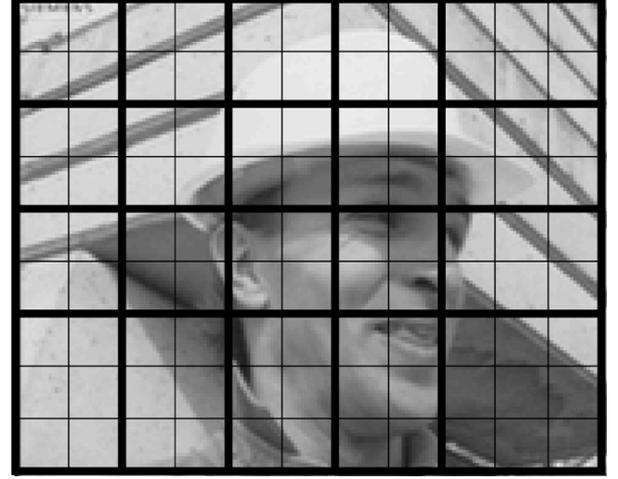


Fig. 4. Image partitioning of a QCIF picture of the sequence *Foreman* into $N = 20$ clusters.

The distortion $D_{DFD}(\mathbf{S}_k, \mathbf{m})$ for the 16×16 block \mathbf{S}_k that is measured between the current picture $s[x, y, t]$ and the decoded reference picture $\hat{s}[x, y, t - \mathbf{m}_t]$ is computed using the sum of squared differences (SSD), while $R(\mathbf{S}_k, \mathbf{m})$ is the number of bits associated with the motion vector. The minimization proceeds over the search space $\mathcal{M} = [-16 \dots 16] \times [-16 \dots 16] \times [0 \dots K - 1]$. First, the integer-sample motion vectors are determined that minimize the Lagrangian cost term in (4) for each of the K reference pictures. Then, these K integer-sample accurate motion vectors are used as initialization of a half-sample refinement step which tests the eight surrounding half-sample positions. Finally, the motion vector among the K candidates is determined as \mathbf{m}^I which minimizes the Lagrangian cost term in (4). Following [1], the Lagrange multiplier is chosen as $\lambda = 0.85 \cdot Q^2$, with Q being the DCT quantizer value, i.e., half the quantizer step size [7].

2) *Affine Refinement:* For the affine refinement step, the initial translational motion vector $\mathbf{m}^I = (\mathbf{m}_x^I, \mathbf{m}_y^I, \mathbf{m}_t^I)$ which is either obtained via the *cluster-based* or *macroblock-based initialization* is used to motion-compensate the past decoded picture $\hat{s}[x, y, t - \mathbf{m}_t]$ toward the current picture $s[x, y, t]$ as follows:

$$\hat{s}[x, y, t] = \hat{s}[x - \mathbf{m}_x^I, y - \mathbf{m}_y^I, t - \mathbf{m}_t^I]. \quad (5)$$

This motion compensation is conducted only for the samples inside the considered cluster \mathcal{A} . The minimization criterion for the affine refinement step reads as follows:

$$\mathbf{a}^R = \arg \min_{\mathbf{a}} \sum_{x, y \in \mathcal{A}} u^2[x, y, t, \mathbf{a}] \quad (6)$$

with

$$u[x, y, t, \mathbf{a}] = s[x, y, t] - \hat{s}[x - \mathbf{m}_x[\mathbf{a}, x, y], y - \mathbf{m}_y[\mathbf{a}, x, y], t] \quad (7)$$

where $\mathbf{m}_x[\mathbf{a}, x, y]$ and $\mathbf{m}_y[\mathbf{a}, x, y]$ have been given in (1).

The signal $\hat{s}[x - \mathbf{m}_x[\mathbf{a}, x, y], y - \mathbf{m}_y[\mathbf{a}, x, y], t]$ is linearized around the spatial location (x, y) for small spatial displacements $(\mathbf{m}_x[\mathbf{a}, x, y], \mathbf{m}_y[\mathbf{a}, x, y])$ yielding

$$\hat{s}[x - \mathbf{m}_x[\mathbf{a}, x, y], y - \mathbf{m}_y[\mathbf{a}, x, y], t] \approx \hat{s}[x, y, t] - \frac{\partial \hat{s}[x, y, t]}{\partial x} \mathbf{m}_x[\mathbf{a}, x, y] - \frac{\partial \hat{s}[x, y, t]}{\partial y} \mathbf{m}_y[\mathbf{a}, x, y]. \quad (8)$$

Hence, the error signal in (7) reads

$$u[x, y, t, \mathbf{a}] \approx s[x, y, t] - \hat{s}[x, y, t] + \frac{\partial \hat{s}[x, y, t]}{\partial x} \mathbf{m}_x[\mathbf{a}, x, y] + \frac{\partial \hat{s}[x, y, t]}{\partial y} \mathbf{m}_y[\mathbf{a}, x, y]. \quad (9)$$

Plugging (1) into (9) and rearranging leads to the following linear equation with six unknowns:

$$u[x, y, t, \mathbf{a}] \approx s[x, y, t] - \hat{s}[x, y, t] + (g_x c_1, g_x c_2 x', g_x c_3 y', g_y c_1, g_y c_2 x', g_y c_3 y') \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} \quad (10)$$

with the abbreviations

$$g_x = \left(\frac{w-1}{2} \right) \frac{\partial \hat{s}[x, y, t]}{\partial x}, \quad g_y = \left(\frac{h-1}{2} \right) \frac{\partial \hat{s}[x, y, t]}{\partial y}, \\ x' = \left(x - \frac{w-1}{2} \right), \quad y' = \left(y - \frac{h-1}{2} \right). \quad (11)$$

Setting up this equation at each sample position inside the cluster leads to an overdetermined set of linear equations that is solved so as to minimize the average squared motion-compensated picture difference. In this work, the pseudoinverse technique is used which is implemented via singular value decomposition. The linearization in (8) holds only for small displacements, which might require an iterative approach to solve (10). However, due to the translational initialization and the subsequent quantization of the affine motion parameters, it turns out that no iteration is needed. Experiments in which the number of iterations have been varied without observing a significant difference in the resulting rate-distortion performance verify this statement.

The spatial intensity gradients are computed following [34] and [35]. With $z \in \{x, y\}$, the spatial gradients are given as

$$\frac{\partial \hat{s}[x, y, t]}{\partial z} = \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 \alpha_{ij}^z s[x+i, y+j, t] + \beta_{ij}^z \hat{s}[x+i, y+j, t], \quad (12)$$

where α_{ij}^z and β_{ij}^z are the element on the i th row and j th column of the matrices

$$\mathbf{A}^x = \mathbf{B}^x = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad \mathbf{A}^y = \mathbf{B}^y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}. \quad (13)$$

The estimates provide the gradient of the point in-between the four samples and between the precompensated and the current image [35]. Since the spatial gradients are computed between the sample positions, the picture difference $s[x, y, t] - \hat{s}[x, y, t]$ is also computed using the summation on the right-hand side of (12) with $z = t$ and

$$\mathbf{A}^t = -\mathbf{B}^t = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (14)$$

The affine motion parameters for motion compensation between the reference picture $\hat{s}[x, y, t - \mathbf{m}_t^I]$ and the current picture $s[x, y, t]$ are obtained via concatenating the initial translational motion vector \mathbf{m}^I and the estimated affine motion parameter set \mathbf{a}^R yielding

$$a_1 = \frac{2\mathbf{m}_x^I}{c_1(w-1)} + a_1^R, \quad a_2 = a_2^R, \quad a_3 = a_3^R \\ a_4 = \frac{2\mathbf{m}_y^I}{c_1(h-1)} + a_4^R, \quad a_5 = a_5^R, \quad a_6 = a_6^R. \quad (15)$$

The initial translational block matching and the affine refinement procedure are repeated for each of the L candidates. As mentioned above, finally, the affine motion parameter set is chosen that minimizes the MSE measured over the samples in the cluster \mathcal{A} .

B. Step B: Reference Picture Warping

For each of the N estimated affine motion parameter sets, the corresponding reference picture is warped toward the current picture. The reference picture warping is conducted using the motion field that is computed via (1) given each affine motion parameter set for the complete picture. Intensity values that correspond to noninteger displacements are computed using cubic spline interpolation [36] which turns out to be more efficient than bilinear interpolation as the motion model becomes more sophisticated [37]. Hence, the multipicture buffer is extended by N new reference pictures that can be used for block-based prediction of the current picture as illustrated in Fig. 1.

C. Step C: Rate-Constrained Multipicture Hybrid Video Encoding

At this point, it is important to note that the multipicture buffer is filled with the K most recent decoded pictures and N warped pictures yielding a total of M reference pictures. Our goal is to obtain a coded representation of the video picture that is efficient in terms of rate-distortion performance via choosing a combination of motion vectors, macroblock modes, and reference pictures. Since affine multipicture MCP is integrated into a hybrid video codec that is based on H.263, we adapt the recommended encoding strategy TMN-10 [38] of H.263 toward the new motion compensation approach.

The TMN-10 encoding strategy as in [38] utilizes macroblock mode decision similar to [4]. For each macroblock, the coding mode with associated parameters is optimized given the decisions made for prior coded blocks only. Let the Lagrange parameter λ and the DCT quantizer value Q be given. The Lagrangian

mode decision for a macroblock \mathcal{S}_k in TMN-10 proceeds by minimizing

$$J_{\text{MODE}}(\mathcal{S}_k, I_k | Q, \lambda) = D_{\text{REC}}(\mathcal{S}_k, I_k | Q) + \lambda \cdot R_{\text{REC}}(\mathcal{S}_k, I_k | Q), \quad (16)$$

where the macroblock mode I_k is varied over the set {INTRA, SKIP, INTER}. Rate $R_{\text{REC}}(\mathcal{S}_k, I_k | Q)$ and distortion $D_{\text{REC}}(\mathcal{S}_k, I_k | Q)$ for the various modes are computed as follows.

For the INTRA mode, the 8×8 blocks of the macroblock \mathcal{S}_k are processed by a DCT and subsequent quantization. Distortion $D_{\text{REC}}(\mathcal{S}_k, \text{INTRA} | Q)$ is measured as the SSD between the reconstructed and the original macroblock samples and $R_{\text{REC}}(\mathcal{S}_k, \text{INTRA} | Q)$ is the rate that results after run-level variable-length coding.

For the SKIP mode, distortion $D_{\text{REC}}(\mathcal{S}_k, \text{SKIP})$ and rate $R_{\text{REC}}(\mathcal{S}_k, \text{SKIP})$ do not depend on the DCT quantizer value Q of the current picture. Distortion is determined as the SSD between the current picture and each of the $M = K + N$ reference pictures for the macroblock samples, and rate is given as one bit per macroblock plus the number of bits necessary to signal the corresponding reference picture. Finally, a reference picture is chosen, for which the SKIP mode provides the smallest cost when evaluating (16).

The computation of the Lagrangian costs for the INTER coding mode is much more demanding than for INTRA and SKIP. This is because of the block motion estimation and motion compensation step. In order to produce the MCP signal, multipicture block-based motion compensation is conducted. That is, half-sample accurate motion vectors $\mathbf{m} = (\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_t)^T$ are applied to compensate blocks of size 16×16 samples referencing one of the $M = K + N$ reference pictures. Again, block-based motion estimation is conducted to obtain the motion vectors by minimizing (4) as it was done when searching decoded pictures to initialize affine motion estimation. In case the *macroblock-based initialization* is employed, the corresponding motion vectors can be re-used. Otherwise, motion estimation over the K decoded pictures has to be conducted as described for the *macroblock-based initialization*. When searching a warped reference picture, only a range of $[-2 \dots 2] \times [-2 \dots 2]$ spatially displaced samples is considered. This small search range is justified by the fact that the warped pictures are already motion-compensated and experiments with a larger search range show that only a very small percentage of motion vectors is found outside the $[-2 \dots 2] \times [-2 \dots 2]$ range. The resulting prediction error signal is similar to the INTRA mode processed by a DCT and subsequent quantization. The distortion D_{REC} is also measured as the SSD between the reconstructed and the original macroblock samples. The rate R_{REC} is given as the sum of the bits for the motion vector and the bits for the quantized and run-level variable-length encoded DCT coefficients.

Finally, the best coding mode, i.e., the one that minimized the Lagrangian cost function in (16), is chosen for each macroblock. During the minimization, the values that correspond to the best coding mode for a given reference pictures are stored in an array. This is done to permit fast access to the Lagrangian costs for the following step, where the number of efficient reference pictures is determined.

D. Step D: Determination of the Number of Efficient Reference Pictures

As mentioned before, there is still the open issue about how to determine an efficient combination of motion vectors, macroblock modes and reference pictures to code the current picture. Because of the interdependency of the various parameters, a locally optimal solution is searched using the precomputed Lagrangian costs from Step C. The greedy optimization algorithm proceeds as follows.

- 1) Sort the $M = K + N$ reference pictures according to the frequency of their selection.
- 2) Starting with the least popular reference picture, evaluate the efficiency of each reference picture by the following two steps.
 - a) For each block that is motion-compensated using the evaluated reference picture, compute its best replacement among the more popular reference pictures in terms of rate-distortion costs.
 - b) If the costs for coding the warping parameters associated with the evaluated reference picture exceed the cost of using the replacement reference pictures, then remove the evaluated reference picture, otherwise keep it.

The first step is conducted because of the use of the variable length code to index the reference pictures. The chosen reference picture with associated warping parameters are transmitted in the header of each picture. The order of their transmission provides the corresponding index that is used to specify a particular reference picture using the block-based motion vectors. This index is entropy-coded using a variable length code and the sorting matches the selection statistics to the length of the code words.

In the second step, the utility of each reference picture is tested by evaluating the rate-distortion improvement obtained by removing this reference picture. For those blocks that reference the removed picture, the best replacements in terms of Lagrangian costs among the more popular reference pictures are selected. Only the more popular pictures are considered because they potentially correspond to a smaller rate and because of the goal to obtain a reduced number of reference pictures in the end. If no rate-distortion improvement is observed, the picture is kept in the reference picture buffer and the procedure is repeated for the next reference picture.

After having determined the number of efficient pictures M^* in the multiple reference picture buffer, the rate-distortion costs of the INTER-4V macroblock mode are also considered and the selected parameters are encoded. Up to this point, the INTER-4V mode has been intentionally left out of the encoding because of the associated complexity to determine the mode costs.

IV. EXPERIMENTS

Within the framework of the multipicture affine motion coder, there are various free parameters that can be adjusted. In this section, empirical justifications are given for important parameter choices made. Attention is given to parameters that have

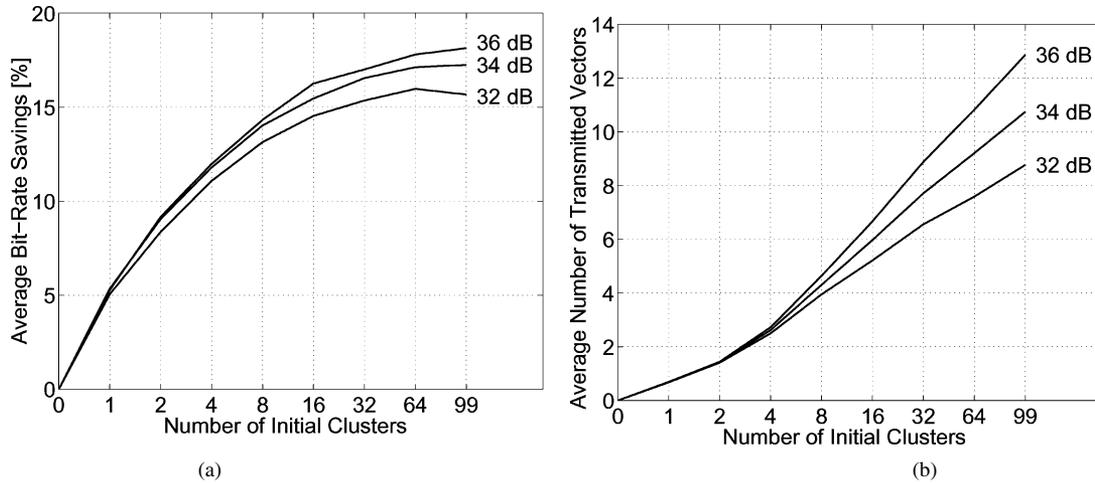


Fig. 5. (a) Average bit-rate savings against TMN-10 and (b) average number of transmitted affine motion parameter sets versus the number of initial clusters for the test sequences in Table I and three different levels of reproduction quality.

the largest impact on the tradeoff between rate-distortion performance and computational complexity. Regarding the affine motion coder, the important question about the number of initial clusters N is discussed. This parameter is very critical since the number of warped reference pictures is directly affected by N . Then, the combination of long-term memory motion compensation with affine motion compensation is investigated and the results when combining them are presented.

A. Affine Motion Compensation

In this section, the parameter setting for the affine motion coder is investigated. For that, the warping is restricted to exclusively reference the prior decoded picture. As shown later, the results for this case also propagate to a setting where the affine motion coder is combined with long-term memory MCP.

The first issue to clarify concerns the number of initial clusters N . For that, the translational motion vector estimation is conducted using the *cluster-based initialization* as described in Section III-A1. The coder is initialized with $N = 1, 2, 4, 8, 16, 32, 64,$ and 99 clusters. The partition into the N initial clusters is conducted so as to obtain equal size blocks and each of the blocks being as close as possible to a square. The translational motion vectors serve as an initialization to the affine refinement step as described in Section III-A2. The estimated affine motion parameter sets are used to warp the previous decoded picture N times as explained in Section III-B. Block-based multipicture motion estimation and determination of the number of efficient affine motion parameter sets is conducted as described in Sections III-C and III-D.

The left-hand side of Fig. 5 shows the average bit-rate savings for the set of test sequences summarized in Table I. For comparison, rate-distortion curves have been generated and the bit rate is measured at equal peak SNR (PSNR). The intermediate points of the rate-distortion curves are interpolated, which allows us to determine the bit rate that corresponds to a particular PSNR value. The percentage in bit-rate savings corresponds to different absolute bit-rate values for the various sequences. Hence, rate-distortion curves are also shown later. Nevertheless, computing bit-rate savings might provide a meaningful measure, for

TABLE I
TEST SEQUENCES AND SIMULATION CONDITIONS

Sequence Name	Abbreviation	Number of Pictures	Picture Skip	Global Motion
<i>Foreman</i>	fm	400	2	Yes
<i>Mobile & Calendar</i>	mc	300	2	Yes
<i>Stefan</i>	st	300	2	Yes
<i>Tempete</i>	te	260	1	Yes
<i>Container Ship</i>	cs	300	2	No
<i>Mother & Daughter</i>	md	300	2	No
<i>News</i>	nw	300	2	No
<i>Silent Voice</i>	si	300	1	No

example, for video content providers who want to guarantee a certain quality of the reconstructed sequences.

The average bit-rate savings against TMN-10 are very similar for the three different levels of reproduction quality. The number of initial clusters has a significant impact on resulting rate-distortion performance. The increase in bit-rate savings saturates for a large number of clusters, i.e., more than 32 clusters, reaching the value of 17% for the set of test sequences considering the reproduction quality of 34-dB PSNR.

This can be explained when investigating the average number of transmitted affine motion parameter sets as shown on the right-hand side of Fig. 5. The curves for the average number of transmitted affine motion parameter sets are generated with a similar method as the average bit-rate savings for a given PSNR value. The average number of affine motion parameter sets increases with increasing average PSNR as well as an increased number of initial clusters. This is because the size of the measurement window becomes smaller as the number of initial clusters increases and the affine motion parameters are more accurate inside the measurement window. Hence, the coder chooses to transmit more affine motion parameter sets. For very small numbers of initial clusters, a large percentage of the maximum number of affine motion parameter sets is chosen. However, as the number of initial clusters is increased, a decreasing percentage of affine motion parameter sets relative to the number of initial clusters is transmitted.

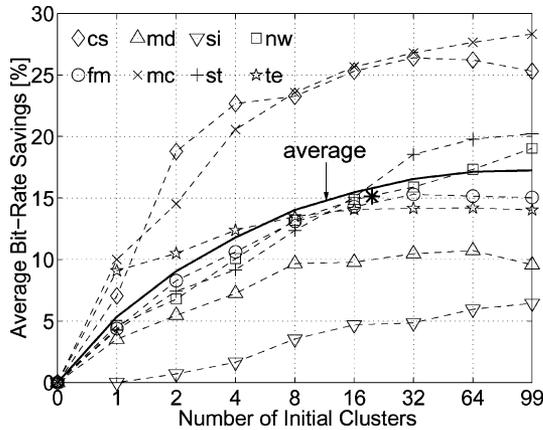


Fig. 6. Average bit-rate savings against TMN-10 at 34-dB PSNR versus the number of initial clusters for the test sequences in Table I. For these results, only the prior coded picture is warped.

Fig. 6 shows the average bit-rate savings against TMN-10 at a reproduction quality of 34-dB PSNR for the set of test sequences where the result for each sequence is shown. The abbreviations *fm*, *mc*, *st*, *te*, *cs*, *md*, *nw*, and *si* correspond to those in Table I. The solid line depicts the average bit-rate savings for the 8 test sequences at equal PSNRs of 34 dB. The results differ quite significantly among the sequences in the test set. On the one hand, for the sequence *Silent Voice*, only a bit-rate saving of 6% can be obtained. On the other hand, sequences like *Mobile & Calendar* and *Container Ship* show substantial gains of more than 25% in bit-rate savings.

In Fig. 6, the asterisk shows the average result for the *macroblock-based initialization* of the affine estimation (see Section III-A). Please recall that all experiments that were described so far are conducted using the *cluster-based initialization* for the translational motion vector estimation to have a simple means for varying the number of initial clusters. For the *macroblock-based initialization*, the segmentation in Fig. 4 is employed resulting in $N = 20$ clusters. The bit-rate saving of 15% is very close to the best result for the *cluster-based initialization*. However, the complexity is drastically reduced.

Typical run-time numbers for the *macroblock-based initialization* are as follows. The complete affine motion coder runs at 6.5 s per QCIF picture on a 300-MHz Pentium PC. These 6.5 s are split into 0.5 s for translational motion estimation for 16×16 macroblocks, 1 s for affine motion estimation, and the warping also takes 1 s. The pre-computation of the costs for the INTER, SKIP, and INTRA mode takes 2 s, and the remaining steps use 2 s. As a comparison, the TMN-10 coder which has a similar degree of run-time optimization uses 2 s per QCIF picture.

Finally, rate-distortion curves are depicted to evaluate the performance of this approach. For that, the DCT quantization parameter has been varied over values $Q = 4, 5, 7, 10, 15$, and 25 when encoding the sequences *Foreman*, *Mobile & Calendar*, *News*, and *Tempete*. The results are shown in Fig. 7, where the rate-distortion curves for the affine motion coder are compared to those of TMN-10 when running both codecs according to the conditions in Table I. The following abbreviations indicate the two codecs compared:

- **TMN-10:** The H.263 test model using Annexes D, F, I, J, and T.
- **MRPW:** As TMN-10, but motion compensation is extended to referencing warped pictures corresponding to $N = 20$ initial clusters using the *macroblock-based initialization*.

The PSNR gains vary for the different test sequences and tend to be larger as the bit rate increases. In contrast, the relative bit-rate savings are more or less constant over the entire range of bit rates that was tested. Typically, a PSNR gain of 1 dB compared to TMN-10 is obtained. The PSNR gains are up to 2.3 dB for the sequence *Mobile & Calendar*.

B. Combination of Affine and Long-Term Memory Motion Compensation

In the previous section, it is shown that affine motion compensation provides significant bit-rate savings against TMN-10. The gains for the affine motion coder increase with an increasing number of initial clusters. A saturation of the gains is reported when increasing the number of initial clusters beyond 32. The number of initial clusters determines the number of reference pictures that are warped. Hence, a parameter choice is proposed where 20 initial clusters are utilized providing an average bit-rate saving of 15%.

In contrast to the affine motion coder, where warped versions of the prior decoded picture are employed, the long-term memory MCP coder references past decoded pictures for motion compensation. However, aside from the different origin of the various reference pictures, the syntax for both codecs is very similar. In [6], the average bit-rate savings against TMN-10 at 34 dB PSNR for the set of test sequences that are achieved with the long-term memory MCP codec are presented. We achieve an average bit-rate reduction of 17% when utilizing 99 additional reference pictures in our long-term memory coder. The bit-rate savings saturate as we further increase the number of reference pictures. Already when utilizing nine additional reference pictures, i.e., using $K = 10$ reference pictures overall, we get 13.8% average bit-rate savings against TMN-10. In [6], it is found that long-term memory MCP with 10 past decoded pictures for most sequences yields a good compromise between complexity and bit-rate savings. Hence, we will use 10 decoded pictures when combining long-term memory prediction and affine motion compensation.

In Fig. 8, the result is depicted when combining the affine motion coder and long-term memory MCP. This plot shows average bit-rate savings against TMN-10 at 34-dB PSNR versus the number of initial clusters for the set of test sequences in Table I. Two cases are shown:

- 1) affine warping using $K = 1$ reference picture (lower solid curve);
- 2) affine warping using $K = 10$ reference pictures (upper solid curve).

For the case $K = 1$, the same setting of the coder is employed that is used for the curve depicting the average bit-rate savings at 34 dB on the left-hand side in Fig. 5. To obtain the result for the case $K = 10$, the combined coder is run using the *cluster-based*

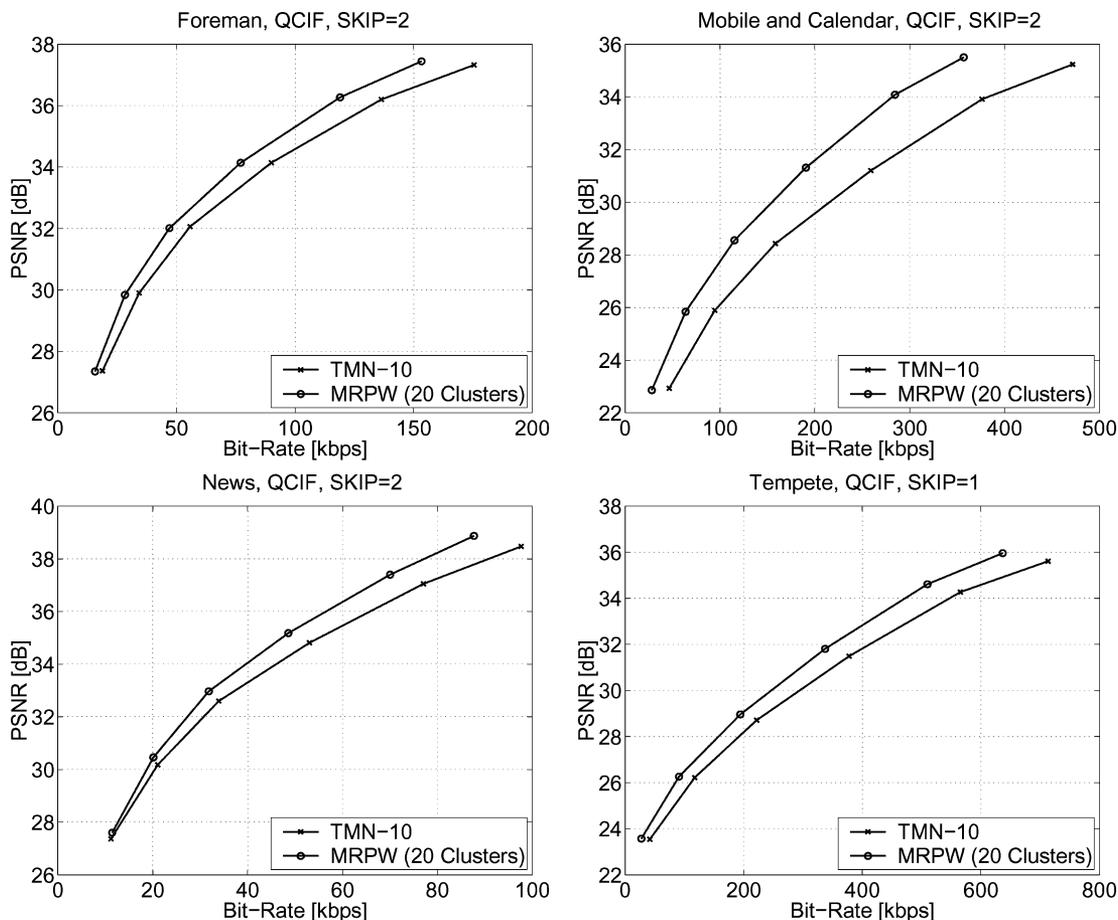


Fig. 7. PSNR versus overall bit rate for the QCIF sequences *Foreman* (top left), *Mobile & Calendar* (top right), *News* (bottom left), and *Tempete* (bottom right).

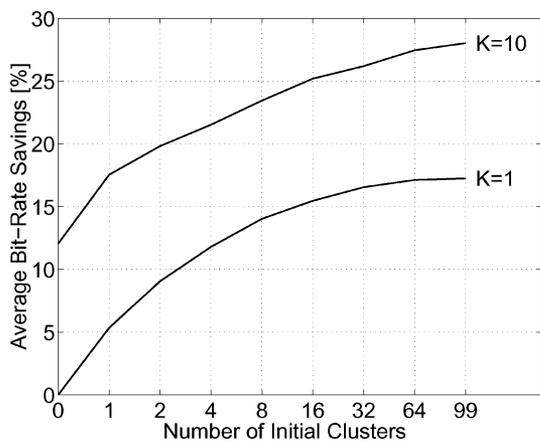


Fig. 8. Average bit-rate savings against TMN-10 at 34-dB PSNR versus the number of initial clusters for the set of test sequences in Table I. Two cases are shown: 1) affine warping using $K = 1$ reference picture (lower solid curve) and 2) affine warping using $K = 10$ reference pictures (upper solid curve).

initialization with $N = 1, 2, 4, 8, 16, 32, 64,$ and 99 initial clusters. For the *cluster-based initialization* of the affine motion estimation, $L = K = 10$ initial translational motion vectors are utilized each corresponding to the best match on one of the K decoded pictures (see Section III-A). Please note that the number of maximally used reference pictures is $N + K$. Interestingly, the average bit-rate savings obtained by the affine motion and the

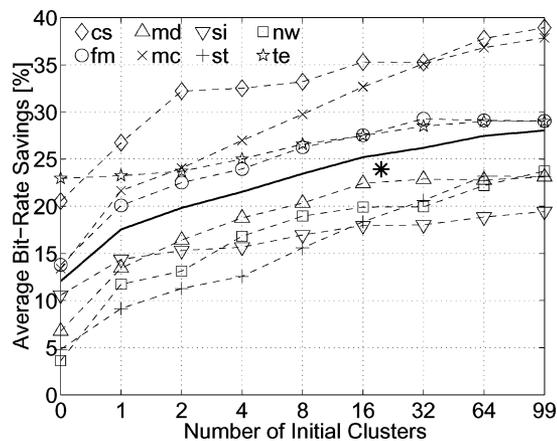


Fig. 9. Average bit-rate savings against TMN-10 at 34 dB PSNR versus number of initial clusters for the set of test sequences in Table I. For these results, the $K = 10$ reference pictures may be utilized for warping.

long-term memory prediction coder are almost additive when being combined using multipicture affine MCP.

Fig. 9 shows the bit-rate savings against TMN-10 for each of the test sequences in Table I when employing $K = 10$ reference pictures versus the number of initial clusters N using dashed lines. The bit-rate savings are more than 35% for the sequences *Container Ship* and *Mobile & Calendar* when using 32 or more

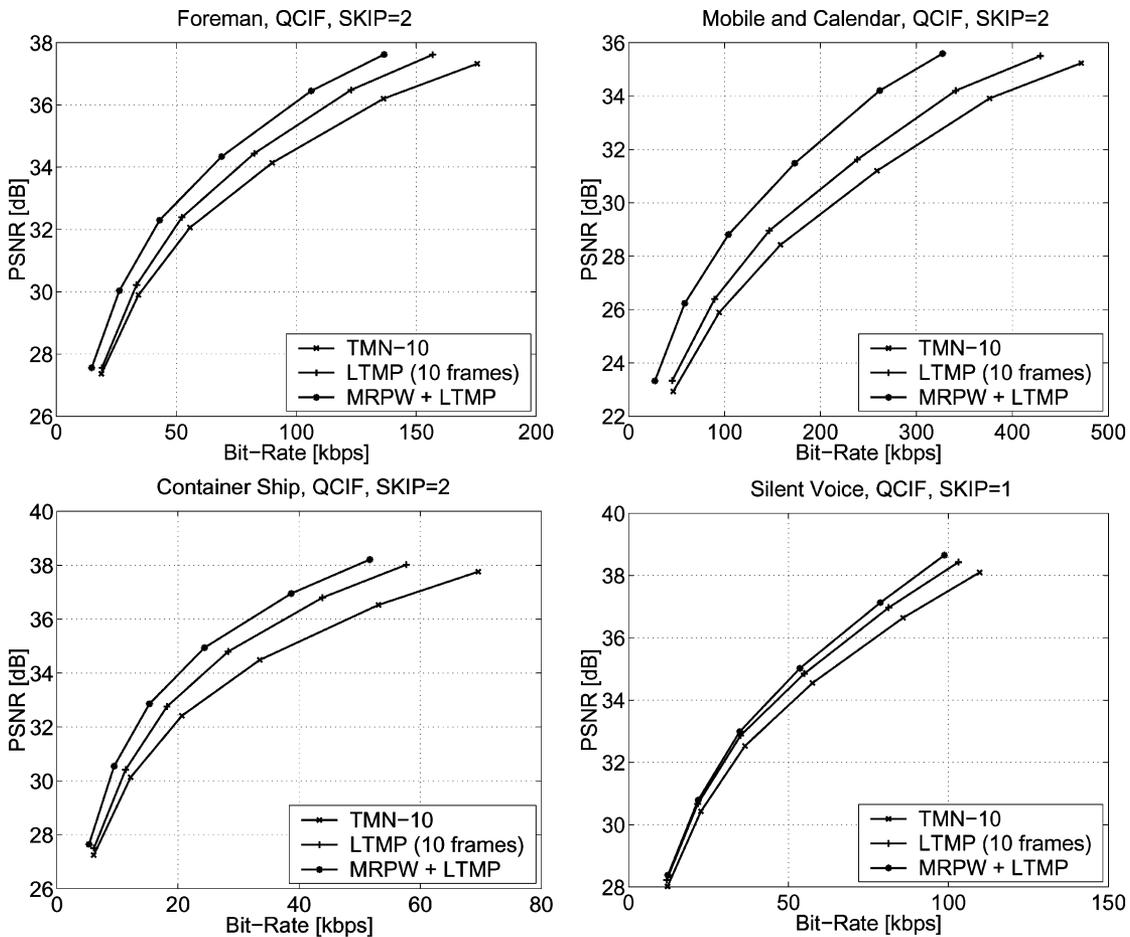


Fig. 10. PSNR versus overall bit rate for the QCIF sequences *Foreman* (top left), *Mobile & Calendar* (top right), *Container Ship* (bottom left), and *Silent Voice* (bottom right).

initial clusters. Interestingly, when using $K = 10$ reference pictures and 16 or more initial clusters the bit-rate savings are never below 17%.

In Fig. 9, the asterisk shows the result for the case of *macroblock-based initialization*. For that, the initial segmentation in Fig. 4 is employed. The initial motion vectors for the affine motion estimation are those best matches found for the macroblocks in each cluster when searching $K = 10$ decoded reference pictures. An average bit-rate saving of 24% is obtained for the set of eight test sequences in Table I.

The measured bit-rate savings correspond to PSNR gains of up to 3 dB. Fig. 10 shows rate-distortion curves for the four test sequences *Foreman*, *Mobile & Calendar*, *Container Ship*, and *Silent Voice*. The curves depict the results that are obtained with the following three coders:

- **TMN-10:** The H.263 test model using Annexes D, F, I, J, and T.
- **LTMP:** As TMN-10, but motion compensation is extended to long-term memory prediction with $K = 10$ decoded reference pictures.
- **MRPW+LTMP:** As TMN-10, but motion compensation is extended to combined affine and long-term memory prediction. The size of the long-term memory is selected as $K = 10$ pictures. The number of initial clusters is $N = 20$ and the *macroblock-based initialization* is employed.

Long-term memory MCP with $K = 10$ pictures and without affine warping is always better than TMN-10 as already demonstrated in [5], [6]. Moreover, long-term memory MCP in combination with affine warping is always better than the case without affine warping. Typically, bit-rate savings between 20 and 35% can be obtained which correspond to PSNR gains of 2–3 dB. For some sequences, long-term memory prediction provides the most gain (*Silent Voice*), while for other sequences the affine motion coder is more important (*Mobile & Calendar*).

For the sequence *Mobile & Calendar*, the gap between the result for the long-term memory MCP codec with and without affine motion compensation is visible for the lowest bit rates as well. This results in a bit-rate saving of 50%. Moreover, for some sequences, the gain obtained by the combined coder is larger than the added gains of the two separate coders. For example, the long-term memory prediction gain for *Mother & Daughter* is 7% for $K = 10$ reference pictures when measuring over all coded pictures. The gain obtained for the affine motion coder is 10% when using 32 initial clusters. However, the combined coder achieves 23% bit-rate savings.

V. CONCLUSION

The idea of reference picture warping can be regarded as an alternative approach to assigning affine motion parameters to large image segments with the aim of a rate-distortion efficient

motion representation. Although the affine motion parameter sets are determined on subareas of the image, they can be employed at any position inside the picture. Instead of performing a joint estimation of the image partition and the associated affine motion parameter sets, reference pictures are warped and selected in a rate-distortion efficient way on a block basis. Hence, the presented approach decomposes the joint optimization task of finding an efficient combination of affine motion parameters, regions and other parameters into separate steps. Each of these steps takes an almost constant amount of computation time which is independent of the context of the input data. The coder robustly adapts the number of affine motion parameter sets to the input statistics and never degrades below the rate-distortion performance that can be achieved with the syntax of the underlying H.263 standard. The use of multiple reference pictures requires only very minor syntax changes to video coding algorithms or is already present as in Annex U of H.263 [7] or in H.264/AVC [8].

The combined affine and long-term memory MCP codec is an example for an efficient multipicture video compression scheme. The two incorporated multipicture concepts seem to complement each other well providing almost additive rate-distortion gains. When warping the prior decoded picture, average bit-rate savings of 15% against TMN-10 are reported for the case that 20 warped reference pictures are used. For these measurements, reconstruction PSNR is identical to 34 dB for all cases considered. These average bit-rate savings are measured over a set of eight test sequences that represent a large variety of video content. Within the test set, the bit-rate savings vary from 6% to 25%. Long-term memory prediction has been already demonstrated as an efficient means to compress motion video [5], [6]. The efficiency in terms of rate-distortion performance is comparable to that of the affine coder. The combination of the two approaches yields almost additive average gains. When employing 20 warped reference pictures and 10 decoded reference pictures, average bit-rate savings of 24% can be obtained for the set of eight test sequences. The minimal bit-rate savings inside the test set are 15% while the maximal bit-rate savings are reported to be up to 35%. These bit-rate savings correspond to gains in PSNR between 0.8–3 dB.

APPENDIX TEST SEQUENCES

The experiments in this paper are conducted using the QCIF test sequences and conditions in Table I. The sequences and test conditions are almost identical to those that are maintained by the ITU-T Video Coding Experts Group. This set of sequences has been chosen so as to represent a wide variety of statistical dependencies and different types of motion and texture.

The first four sequences contain a large amount of motion including a moving camera position and focal length change. The last four sequences are low motion sequences with a fixed camera. This set was chosen so as to cover a broad range of possible scenes that might occur in applications such as video conferencing or video streaming.

In all experiments, bit streams are generated that are decodable producing the same PSNR values at encoder and decoder.

The first picture of the image sequence is coded in INTRA mode followed by INTER-coded pictures. In INTER pictures, the macroblocks can either be coded predictively using one of the INTER macroblock modes or as INTRA blocks. In the simulations, the first intracoded picture is identical for all cases considered.

REFERENCES

- [1] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [2] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. GLOBECOM'91*, Phoenix, AZ, Dec. 1991, pp. 85–90.
- [3] B. Girod, "Rate-constrained motion estimation," in *Proc. SPIE Conf. Visual Commun. Image Process.*, vol. 2308, Chicago, IL, Sep. 1994, pp. 1026–1034.
- [4] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 182–190, Apr. 1996.
- [5] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 70–84, Feb. 1999.
- [6] T. Wiegand and B. Girod, *Multi-Frame Motion-Compensated Prediction for Video Transmission*. Norwell, MA: Kluwer, 2001.
- [7] "Video Coding for Low Bitrate Communication," ITU-T Recommendation H.263 Version 3 (H.263++), 2000.
- [8] "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264 & ISO/IEC 14 496–10 AVC, 2003.
- [9] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
- [10] R. Y. Tsai and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-29, no. 6, pp. 1147–1152, Dec. 1981.
- [11] M. Hötter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation," *Signal Process.: Image Commun.*, vol. 15, no. 3, pp. 315–334, Oct. 1988.
- [12] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Process.: Image Commun.*, vol. 3, no. 1, pp. 23–56, Jan. 1991.
- [13] H. Sanson, "Motion affine models identification and application to television image sequences," in *Proc. SPIE Conf. Visual Commun. Image Process.*, vol. 1605, 1991, pp. 570–581.
- [14] Y. Yokoyama, Y. Miyamoto, and M. Ohta, "Very low bit rate video coding using arbitrarily shaped region-based motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 12, pp. 500–507, Dec. 1995.
- [15] C. K. Cheong, K. Aizawa, T. Saito, M. Kaneko, and H. Harashima, "Structural motion segmentation for compact image sequence representation," in *Proc. SPIE Conf. Visual Commun. Image Process.*, vol. 2727, Orlando, FL, Mar. 1996, pp. 1152–1163.
- [16] E. Francois, J.-F. Vial, and B. Chupeau, "Coding algorithm with region-based motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 97–108, Feb. 1987.
- [17] S.-C. Han and J. W. Woods, "Adaptive coding of moving objects for very low bit rates," *IEEE J. Select. Areas Commun.*, vol. 16, no. 1, pp. 56–70, Jan. 1998.
- [18] H. Li and R. Forchheimer, "A transform block-based motion compensation technique," *IEEE Trans. Commun.*, vol. 43, no. 2, pp. 1673–1676, Feb. 1995.
- [19] K. Zhang, M. Bober, and J. Kittler, "Image sequence coding using multiple-level segmentation and affine motion estimation," *IEEE J. Select. Areas Commun.*, vol. 15, no. 12, pp. 1704–1713, Dec. 1997.
- [20] M. Karczewicz, J. Nieweglowski, and P. Haavisto, "Video coding using motion compensation with polynomial motion vector fields," *Signal Process.: Image Commun.*, vol. 10, no. 3, pp. 63–91, Jul. 1997.
- [21] F. Dufaux and F. Moscheni, "Background mosaicking for low bit rate video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Lausanne, Switzerland, Sep. 1996, pp. 673–676.
- [22] "Core Experiment on Sprites and GMC," ISO/IEC JTC1/SC29/WG11 MPEG96/N1648, 1997.

- [23] A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 12, pp. 1227–1242, Dec. 1999.
- [24] N. Mukawa and H. Kuroda, "Uncovered background prediction in interframe coding," *IEEE Trans. Commun.*, vol. COM-33, no. 11, pp. 1227–1231, Nov. 1985.
- [25] D. Hepper, "Efficiency analysis and application of uncovered background prediction in a low bit rate image coder," *IEEE Trans. Commun.*, vol. 38, pp. 1578–1584, Sep. 1990.
- [26] X. Yuan, "Hierarchical uncovered background prediction in a low bit-rate video coder," in *Proc. Picture Coding Symp.*, Lausanne, Switzerland, Mar. 1993, p. 12.1.
- [27] K. Zhang and J. Kittler, "A background memory update scheme for H.263 video codec," in *Proc. Eur. Signal Process. Conf.*, vol. 4, Island of Rhodes, Greece, Sep. 1998, pp. 2101–2104.
- [28] "Coding of Audio-Visual Objects – Part 2: Visual," MPEG-4 Visual Version 1, ISO/IEC 14496-2, 1999.
- [29] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 625–638, Sep. 1994.
- [30] M. Hötter, "Differential estimation of the global motion parameters zoom and pan," *Signal Process.*, vol. 16, no. 3, pp. 249–265, Mar. 1989.
- [31] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global MC and local affine MC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 75–85, Feb. 1997.
- [32] "Core Experiment on Global Motion Compensation," Video Subgroup, ISO/IEC JTC1/SC29/WG11 MPEG96/M1686, 1997.
- [33] "Video Codec for Audiovisual Services at $p \times 64$ kbit/s," ITU-T Recommendation H.261, 1993.
- [34] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [35] B. K. P. Horn, *Robot Vision*. Cambridge, MA/New York: MIT Press/McGraw-Hill, 1986.
- [36] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
- [37] J.-L. Dugelay and H. Sanson, "Differential methods for the identification of 2D and 3D motion models in image sequences," *Signal Process.: Image Commun.*, vol. 7, no. 1, pp. 105–127, Mar. 1995.
- [38] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.



Thomas Wiegand (M'05) received the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Nuremberg, Germany, in 2000 and the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Hamburg, Germany, in 1995.

He is currently the head of the Image Communication Group, Image Processing Department, Heinrich Hertz Institute, Berlin, Germany. From 1993 to 1994, he was a Visiting Researcher with Kobe University, Japan. In 1995, he was a Visiting Scholar with the

University of California, Santa Barbara, where he started his research on video compression and transmission. Since then, he has published several conference and journal papers on the subject and has contributed successfully to the ITU-T Video Coding Experts Group (ITU-T SG16 Q.6 – VCEG)/ISO/IEC Moving Pictures Experts Group (ISO/IEC JTC1/SC29/WG11 – MPEG)/Joint Video Team (JVT) standardization efforts and holds various international patents in this field. From 1997 to 1998, he was a Visiting Researcher with Stanford University, Stanford, CA, and served as a consultant to 8×8 , Inc., Santa Clara, CA. In October 2000, he was appointed as the Associated Rapporteur of the ITU-T VCEG. In December 2001, he was appointed as the Associated Rapporteur/Co-Chair of the JVT that was created by ITU-T VCEG and ISO/IEC MPEG for finalization of the H.264/AVC video coding standard. In February 2002, he was appointed as the Editor of the H.264/AVC video coding standard. His research interests include image and video compression, communication and signal processing as well as vision and computer graphics.



Eckeard Steinbach studied electrical engineering at the University of Karlsruhe, Karlsruhe, Germany, the University of Essex, Essex, U.K., and École Supérieure d'Ingénieurs en Électronique et Électrotechnique (ESIEE), Paris, France. He received the Dipl.-Ing. degree from University of Karlsruhe in 1994. He received the Ph.D. degree in engineering from the University of Erlangen-Nuremberg, Nuremberg, Germany, in 1999.

From 1994 to 2000, he was a Member of the Research Staff of the Image Communication Group, University of Erlangen-Nuremberg. From February 2000 to December 2001, he was a Postdoctoral Fellow with the Information Systems Laboratory, Stanford University, Stanford, CA. In February 2002, he joined the Department of Electrical Engineering and Information Technology, Munich University of Technology, Munich, Germany, where he is currently an Associate Professor for Media Technology. His current research interests are in the area of networked and interactive multimedia systems.

Dr. Steinbach served as a conference co-chair of SPIE Visual Communications and Image Processing (VCIP) in San Jose, CA, in 2001 and Vision, Modeling and Visualization 2003 (VMV) in Munich, November 2003. He has been a Guest Editor of the Special Issue on Multimedia over IP and Wireless Networks of the *EURASIP Journal on Applied Signal Processing*.



Bernd Girod (S'80–M'80–SM'97–F'98) received the M.S. degree in electrical engineering from Georgia Institute of Technology, Atlanta, in 1980 and the Ph.D. degree (with highest honors) from the University of Hannover, Hannover, Germany, in 1987.

He is presently a Professor of Electrical Engineering with the Information Systems Laboratory, Stanford University, Stanford, CA. He also holds a courtesy appointment with the Stanford Department of Computer Science and he serves as Director of the Image Systems Engineering Program at Stanford. His research interests include networked media systems, video signal compression and coding, and three-dimensional image analysis and synthesis. Until 1987, he was a Member of the Research Staff with 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 Massachusetts Institute of Technology, Cambridge, first as a Visiting Scientist with the Research Laboratory of Electronics and then as an Assistant Professor of Media Technology at the Media Laboratory. From 1990 to 1993, he was a Professor of Computer Graphics and Technical Director of the Academy of Media Arts, Cologne, Germany, jointly appointed with the Computer Science Section of Cologne University. He was a Visiting Adjunct Professor with the Digital Signal Processing Group, Georgia Institute of Technology, Atlanta, in 1993. From 1993 until 1999, he was the Chaired Professor of Electrical Engineering/Telecommunications, University of Erlangen-Nuremberg, Nuremberg, Germany, and the Head of the Telecommunications Institute I, codirecting the Telecommunications Laboratory. He has served as the Chairman of the Electrical Engineering Department from 1995 to 1997 and as Director of the Center of Excellence "3-D Image Analysis and Synthesis" from 1995 to 1999. He was a Visiting Professor with the Information Systems Laboratory of Stanford University during the 1997–1998 academic year. As an entrepreneur, he has worked successfully with several start-up ventures as founder, investor, director, or advisor. Most notably, he has been a cofounder and Chief Scientist of Vivo Software, Inc., Waltham, MA (1993–1998); after Vivo's acquisition, Chief Scientist of RealNetworks, Inc. (1998–2002), and an outside Director of 8×8 , Inc. (1996–2004). He has authored or coauthored one major textbook, two monographs, and over 250 book chapters, journal articles, and conference papers in his field, and he holds about 20 international patents.

Prof. Girod has been a member of the IEEE Image and Multidimensional Signal Processing Committee from 1989 to 1997. He was named "Distinguished Lecturer" in 2002 by the IEEE Signal Processing Society. Together with J. Eggers, he was the recipient of the 2002 EURASIP Best Paper Award. He has served on the Editorial Boards or as an Associate Editor for several journals in his field and is currently Area Editor for Speech, Image, Video Signal Processing of the IEEE TRANSACTIONS ON COMMUNICATIONS. He has served on numerous conference committees, e.g., as Tutorial Chair of ICASSP-97 in Munich, Germany, and ICIP-2000 in Vancouver, ON, Canada, as General Chair of the 1998 IEEE Image and Multidimensional Signal Processing Workshop in Alpbach, Austria, and as General Chair of the Visual Communication and Image Processing Conference (VCIP) in San Jose, CA, in 2001.