Model-Aided Coding: A New Approach to Incorporate Facial Animation into Motion-Compensated Video Coding

Peter Eisert, Thomas Wiegand, *Member, IEEE,* and Bernd Girod, *Fellow, IEEE*

***Abstract*—We show that traditional waveform coding and 3-D model-based coding are not competing alternatives, but should be combined to support and complement each other. Both approaches are combined such that the generality of waveform coding and the efficiency of 3-D model-based coding are available where needed. The combination is achieved by providing the block-based video coder with a second reference frame for prediction, which is synthesized by the model-based coder. The model-based coder uses a parameterized 3-D head model, specifying shape and color of a person. We therefore restrict our investigations to typical videotelephony scenarios that show head-and-shoulder scenes. Motion and deformation of the 3-D head model constitute facial expressions which are represented by facial animation parameters (FAP's) based on the MPEG-4 standard. An intensity gradient-based approach that exploits the 3-D model information is used to estimate the FAP's, as well as illumination parameters, that describe changes of the brightness in the scene. Model failures and objects that are not known at the decoder are handled by standard block-based motion-compensated prediction, which is not restricted to a special scene content, but results in lower coding efficiency. A Lagrangian approach is employed to determine the most efficient prediction for each block from either the synthesized model frame or the previous decoded frame. Experiments on five video sequences show that bit-rate savings of about 35% are achieved at equal average peak signal-to-noise ratio (PSNR) when comparing the model-aided codec to TMN-10, the state-of-the-art test model of the H.263 standard. This corresponds to a gain of 2–3 dB in PSNR when encoding at the same average bit rate.**

***Index Terms*—Facial animation, model-aided coding, model-based coding, multiframe prediction.**

I. Introduction

In RECENT years, several video coding standards, such as H.261, MPEG-1, MPEG-2, and H.263 have been introduced to address the compression of digital video for storage and communication services. H.263 [1] as well as the other standards describe a hybrid video coding scheme, which consists of block-based motion-compensated prediction (MCP) and discrete cosine transform (DCT)-based quantization of the prediction error. The future MPEG-4 standard [2] will also follow the same video coding approach. These schemes utilize the statistics of the video signal without knowledge of the semantic content of the frames and can therefore be used for arbitrary scenes.

In case semantic information about a scene is suitably incorporated, higher coding efficiency may result with model-based video codecs [3], [4]. For example, 3-D models that describe the shape and texture of the objects in the scene could be used. The 3-D object descriptions are encoded only once. When encoding a video sequence, individual frames are characterized by 3-D motion and deformation parameters of these objects. In most cases, such parameters can be transmitted at extremely low bit rates. Unfortunately, such a codec lacks generality. It is restricted to scenes that can be composed of objects that are known by the decoder. One typical class of scenes are head-and-shoulder sequences which are frequently encountered in applications such as videotelephony or videoconferencing. For head-and-shoulder scenes, bit rates of about 1 kb/s with acceptable quality can be achieved [5]. This has also motivated the recently determined synthetic and natural hybrid coding (SNHC) part of the MPEG-4 standard [2]. SNHC allows the transmission of a 3-D face model that can be animated to generate different facial expressions.

The transmission of SNHC-based 3-D models is supported in combination with 2-D video streams [2]. The video frame is composited at the decoder out of arbitrarily shaped video object planes (VOP), and each VOP can be either synthesized or conventionally generated by a DCT-based motion-compensated hybrid decoder. Due to the independent encoding of the VOP's and the additional bit rate needed for transmitting their shapes, MPEG-4 SNHC may require a prohibitive amount of overhead information.

Another coding approach that uses multiple compression strategies has been proposed as *dynamic coding* [6]. Choosing from several available compression techniques, the frames are segmented in a rate-distortion optimal sense. Again, the shape information of the regions has to be transmitted as side information and encoding of individual regions is performed independently.

The combination of traditional hybrid video coding methods with model-based coding has been proposed by Chowdhury *et al.* in 1994 [7]. In [7], a *switched model-based coder* is introduced that decides between the encoded output frames from an H.261 coder and a 3-D model-based coder. The frame selection is based on rate and distortion. However, the mode decision is only done for a complete frame and therefore the information from the 3-D model cannot be exploited if parts of the frame cannot be described by the model-based coder.

Manuscript received March 15, 1999; revised September 30, 1999. This paper was recommended by Guest Editor Y. Wang.

The authors are with the Telecommunications Laboratory, University of Erlangen-Nuremberg, D-91058 Erlangen, Germany (e-mail: eisert@nt.e-technik.uni-erlangen.de; wiegand@nt.e-technik.uni-erlangen.de; girod@nt.e-technik.uni-erlangen.de).

Publisher Item Identifier S 1051-8215(00)02793-2.

An extension to the switched model-based coder is the *layered coder* published by Musmann in 1995 [8], as well as Kampmann and Ostermann in 1997 [9]. The layered coder chooses the output from up to five different coders. The mode decision between the layers is also done framewise or objectwise and, again, encoding in the various modes is carried out independently.

In this paper we present an extension of an H.263 video codec [1] that incorporates information from a model-based coder in a novel way. Instead of exclusively predicting the current frame of the video sequence from the previous decoded frame, motion compensated prediction using the synthesized output frame of the model-based coder is also considered. Thus, our codec employs *multiframe prediction* with $M = 2$ frames. In our previous work, we have explored multiframe prediction with up to $M = 50$ frame stores [10]. However, these frame stores either contain past reconstructed frames directly, or warped versions of these [11], [12], but not yet synthetic frames generated with a 3-D model.

With multiframe prediction, the video coder decides which frame should be used for each block by minimizing a Lagrangian cost function $D + \lambda R$, where distortion D is minimized together with rate R. The Lagrange parameter λ controls the balance between distortion and rate. A large value of λ corresponds to low bit rate and large distortion while a small λ results in high bit rate and low distortion [13]–[15]. The minimization proceeds over all available prediction signals and chooses the most efficient one in terms of the cost function.

The incorporation of the model-based coder into the motion-compensated predictor allows the coder to further refine an imperfect model-based prediction. Neither the switched model-based coder [7] nor the layered coder [8], [9] nor MPEG-4 SNHC [2] allow efficient residual coding of the synthesized frame. In contrast, the coding efficiency of our "model-aided codec" (MAC) never degrades below H.263 in the case the model-based coder cannot describe the current scene. However, if the objects in the scene correspond to the 3-D models in the codec, a significant improvement in coding efficiency can be achieved.

This paper is organized as follows. In Section II, we describe the architecture of the video coder that combines the traditional hybrid video coding loop with a model-based coder that is able to encode head-and-shoulder scenes at very low bit rates. In Section III, the underlying semantic model is presented and the algorithm for the estimation of facial animation parameters (FAP's) that determine the rendered output frame is explained. Given the rendered output frame of the model-based coder, we describe how bit allocation is done in our combined block- and model-based coder using Lagrangian optimization techniques (Section IV). Finally, experimental results verify the improved rate-distortion performance of the proposed scheme compared to TMN-10, the test model of the H.263 standard (Section V).

II. Video Coding Architecture

Fig. 1 shows the architecture of the proposed model-aided video coder. It depicts the well-known hybrid video-coding loop that is extended by a model-based codec. The model-based

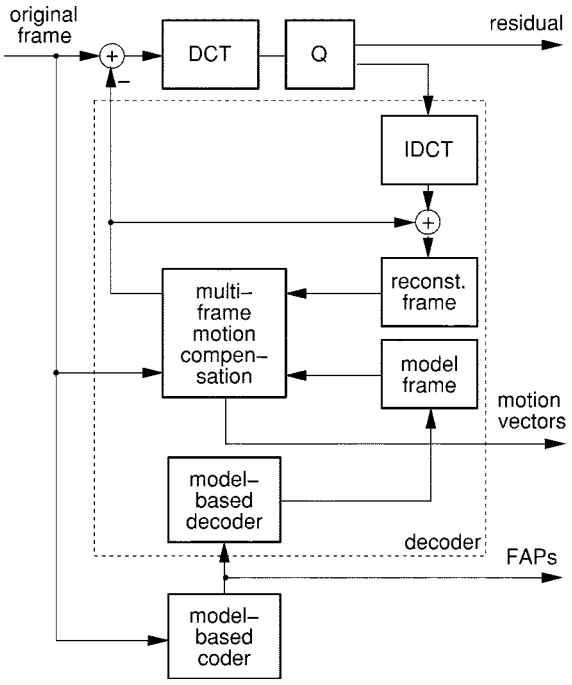


Fig. 1. Structure of the "model-aided coder." Traditional block-based MCP from the previous decoded frame is extended by prediction from the current model frame.

codec runs in parallel to the hybrid video codec, generating a synthetic model frame. This model frame is employed as a second reference frame for block-based MCP in addition to the previous reconstructed reference frame. For each block, the video coder decides which of the two frames to use for MCP. The bit-rate reduction for the proposed scheme compared to single-frame prediction arises from those parts in the image that are well approximated by the model frame. For these blocks, the bit rate required for transmission of the motion vector and DCT coefficients for the residual coding is often dramatically reduced.

Incorporating the model-based codec into an H.263 video codec requires syntax extensions. To enable multiframe MCP, the interprediction macroblock modes INTER and UNCODED are assigned one codeword representing the picture reference parameter for the entire macroblock. The INTER-4V macroblock mode utilizes four picture reference parameters, each associated with one of the four 8×8 block motion vectors. For further details on H.263 syntax, please refer to the ITU-T Recommendation [1].

In addition to the picture reference parameter for each block, the FAP's for synthesizing the model frame at the decoder are included in the picture header. In the next section, we describe the model-based codec in more detail. After that, we return to the combined video codec and explain the coder control.

III. Model-Based Codec

The structure of the model-based codec which is described in the following is depicted in Fig. 2. The coder analyzes the incoming frames and estimates the parameters of the 3-D motion and deformation of the head model. These deformations are represented by a set of FAP's that are quantized, entropy-coded,

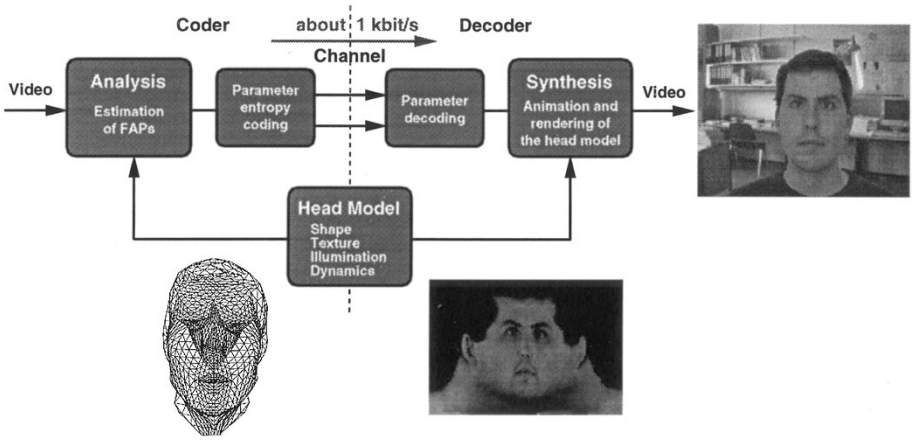


Fig. 2. Basic structure of our model-based codec.

and transmitted through the channel. The 3-D head model with its facial expression synthesis are incorporated into the parameter estimation. The 3-D head model consists of shape, texture, and the description of facial expressions. For synthesis of facial expressions, the transmitted FAP's are used to deform the 3-D head model. Finally, the original video frame is approximated by rendering the 3-D model using standard computer graphics techniques.

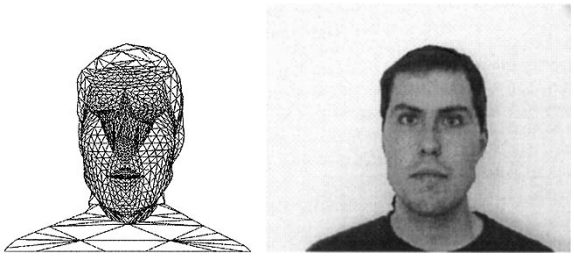
A. 3-D Head Model

For the description of the shape of head and shoulder, we use a generic 3-D model employing a triangle mesh with fixed topology similar to the well-known Candide model [16]. Texture is mapped onto the triangle mesh to obtain photo-realistic appearance. In contrast to the Candide model, we describe the object's surface with triangular B-splines [17] in order to reduce the number of degrees of freedom of the shape. This simplifies modeling and estimation of the facial expressions and does not severely restrict the possible shapes of the surface since facial expressions result in smooth movements of surface points due to the anatomical properties of tissue and muscles. B-splines are well suited to model the surface properties of facial skin, as shown, e.g., by Hoch *et al.* [18].

To reduce the computational complexity of rendering, the B-splines are only evaluated at fixed discrete positions of the surface. These points form the vertices of a triangle mesh that approximates the smooth B-spline surface. The number of vertices, and thus triangles, can be varied to trade approximation accuracy against rendering complexity. The positions of the vertices in 3-D space and the shape of the head are determined in our model by the location of 231 control points of the B-spline surface.

The resulting triangle mesh is colored by mapping texture onto the surface [19]. To each vertex of the mesh, a fixed texture coordinate is assigned that is obtained by projecting the neutral vertex position on a cylindric surface that specifies the texture map.

An example of such a triangle mesh together with the corresponding textured model is shown in Fig. 3.



(a) (b)

Fig. 3. (a) Hidden-line representation of the head model. (b) Corresponding textured version.

B. Synthesis of the Model Frame

The synthesis of the model frame consists of first animating the 3-D head model using the FAP's and then rendering the model frame. The mechanism by which changes of the FAP's influence the shape and mimic of the face in the rendered model frame is defined by a set of cascaded transformations as shown in Fig. 4. Given a set of FAP's defining a certain facial expression, the corresponding 2-D model frame is created by first placing the control points in order to shape the B-spline surface. Using the basis functions of the B splines, the algorithm computes the position of the vertices from the control points. The 3-D location of all object points on the triangle surface is specified by their barycentric coordinates. Finally, the 2-D object points in the model frame are obtained by projecting the 3-D points into the image plane. In the following, all transformations are explained in more detail.

1) Deformation of 3-D Model: We parameterize a person's facial expressions following the proposal of the MPEG-4/SNHC group [2]. According to that scheme, every facial expression can be generated by a superposition of different basic expressions, each quantified by a facial animation parameter (FAP). These FAP's describe elementary motion fields of the face's surface including both global motion, like head rotation, and local motion, like eye or mouth movement. The vector **FAP** that contains all K facial animation parameter values

$$\mathbf{FAP} = [\mathrm{FAP}_0 \;\; \mathrm{FAP}_1 \;\; \cdots \;\; \mathrm{FAP}_k \;\; \cdots \;\; \mathrm{FAP}_{K-1}]^T \qquad (1)$$

then describes the complete facial expression.

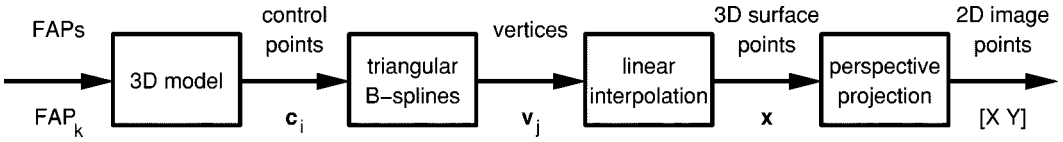


Fig. 4. Transformation from FAP's to image points.

Changes of FAP's influence the shape and facial expression of the 3-D head model. The relation between a parameterized facial expression and the surface shape is defined by a set of transformations that is applied to the control points of the B-spline surface. The final control point position $\boldsymbol{c}_i$ is obtained by concatenating all transformations, each associated with one FAP, according to

$$\boldsymbol{c}_i = T_{\mathrm{FAP}_{K-1}}(\cdots T_{\mathrm{FAP}_1}(T_{\mathrm{FAP}_0}(\boldsymbol{c}_{i0}))) \tag{2}$$

with $\boldsymbol{c}_{i0}$ being the initial control point location corresponding to a neutral expression of the person. Each transformation corresponding to a FAP describes either a translation or a rotation that is quantified by the parameter FAP_k

$$T_{\mathrm{FAP}_k}(\boldsymbol{c}_i) = \begin{cases} \boldsymbol{c}_i + \boldsymbol{d}_{ik} \cdot \mathrm{FAP}_k, & \text{translation} \\ \boldsymbol{R}_{\mathrm{FAP}_k}(\boldsymbol{c}_i - \boldsymbol{o}_k) + \boldsymbol{o}_k, & \text{rotation.} \end{cases} \tag{3}$$

In case of a translation, the control point $\boldsymbol{c}_i$ is moved in direction $\boldsymbol{d}_{ik}$ by the amount of $\|\boldsymbol{d}_{ik}\| \cdot \mathrm{FAP}_k$. For rotational movements, the control point $\boldsymbol{c}_i$ is rotated around $\boldsymbol{o}_k$ as specified by the rotation matrix $\boldsymbol{R}_{\mathrm{FAP}_k}$. $\boldsymbol{R}_{\mathrm{FAP}_k}$ is determined by a fixed rotation axis and a rotation angle proportional to the parameter FAP_k.

To generate facial expressions specified by a set of FAP's, the transformations are computed and sequentially applied on the control point positions. Note that the resulting facial expression is not independent of the order of the transformations. For the rendering of new expressions, we first deform the neutral head model locally and apply the global head rotation and translation last.

2) B-Spline Surface Approximation by Triangle Meshes: Once all control points are properly positioned according to the FAP's, the shape of the B-spline surface is fully determined. To approximate this surface by a triangle mesh, we compute the position of the mesh's vertices. This is accomplished using the linear relation

$$\boldsymbol{v}_j = \sum_{i \in \mathcal{I}_j} b_{ji} \boldsymbol{c}_i \tag{4}$$

between a vertex $\boldsymbol{v}_j$ and the control points $\boldsymbol{c}_i$ with b_{ji} being the basis functions of the B-spline. The basis function values are computed only once and stored in a list. $\mathcal{I}_j$ is the index set that contains the indices of the control points that influence the position of vertex $\boldsymbol{v}_j$. The number of indices in this set is usually between three and six.

From the position of the vertices, arbitrary object points $\boldsymbol{x}$ on the surface of the triangle mesh are computed using

$$\boldsymbol{x} = \sum_{j \in \mathcal{I}} \lambda_j \boldsymbol{v}_j \tag{5}$$

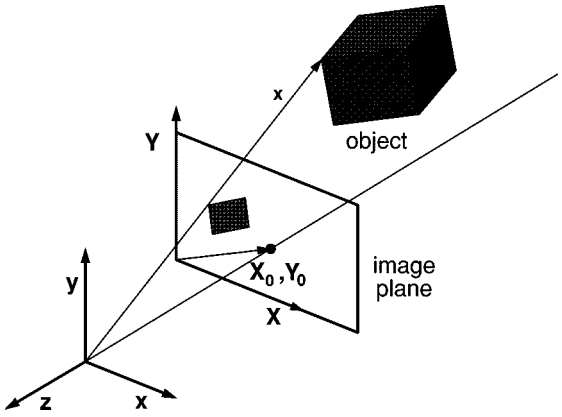


Fig. 5. Camera model and its associated coordinate systems.

where λ_j are the barycentric coordinates of the object point relative to the surface triangle enclosing it. The index set $\mathcal{I}$ contains the three indices of the vertices forming the triangle.

3) Perspective Projection: Given the 3-D object points, a 2-D model frame is rendered using perspective projection. The geometry of the camera model is shown in Fig. 5. The 3-D coordinates of an object point $\boldsymbol{x} = [x\ y\ z]^T$ are projected into the image plane $[X \quad Y]$ according to

$$\begin{aligned} X &= X_0 - f_x \frac{x}{z} \\ Y &= Y_0 - f_y \frac{y}{z} \end{aligned} \tag{6}$$

with f_x and f_y denoting the scaled focal length parameters that allow the use of nonsquare pixel geometries. The two parameters X_0 and Y_0 describe the location of the optical axis and can account for its deviation from the image center due to inaccurate placement of the image sensor in the camera. The four parameters f_x, f_y, X_0, and Y_0 are obtained from an initial camera calibration using Tsai's algorithm [20].

C. Scene Analysis and Encoding of Facial Animation Parameters

The model-based coder analyzes the incoming frames and estimates the 3-D motion and facial expression of a person. The determined FAP's are quantized, entropy coded, and transmitted. The 3-D head model and the motion constraints used by the facial expression synthesis are incorporated into the parameter estimation as described in the following.

1) Facial Parameter Estimation: In our model-based coder all FAP's are estimated simultaneously using a hierarchical optical flow based method [21]. We employ a hierarchy of three spatial resolution layers with CIF as the highest resolution and each subsequent lower resolution layer subsampled by a factor of two, vertically and horizontally. We use the whole picture

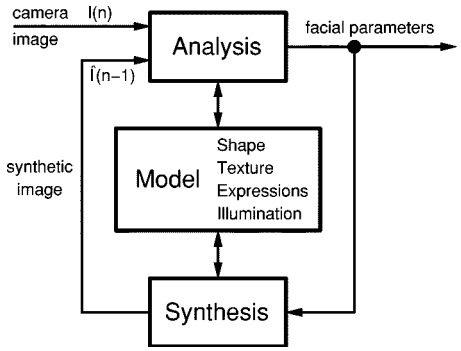


Fig. 6. Analysis-synthesis loop of the coder.

of the face for the estimation, in contrast to feature-based approaches, where only discrete feature point correspondences are exploited. To simplify the optimization in the high-dimensional parameter space, a linearized solution is directly computed from the optical flow with the motion constraints of the head model.

In the optimization, an analysis–synthesis loop is employed [22] as shown in Fig. 6. The algorithm estimates the facial parameter changes between the previous synthetic frame $\hat{I}[n-1]$ and the current frame $I[n]$ from the camera video sequence. This approximate solution is used to compensate the differences between the two frames by rendering the deformed 3-D model at the new position. The remaining linearization errors are reduced by repeating this procedure in the analysis–synthesis loop.

Note that in our experiments the texture map of the head model, once complete, is not updated during the video sequence. All changes between the model image and the video frame must therefore be compensated by the underlying 3-D model, the motion model, and the illumination model. To cope with changes in illumination which have a strong influence on the image intensities we also estimate the current lighting frame-by-frame [23]. The parameters that describe the direction and intensity of the incident light are then used at the decoder to reduce the differences between camera and model frames caused by photometric effects.

In the following, the components of the model parameter estimation algorithm are described in more detail.

2) 3-D Motion Constraint: The 3-D head model restricts the possible deformations in the face to facial expressions that can be controlled by the FAP's. Thus, the motion of any surface point can be described by a function of the unknown FAP changes. This motion constraint can be derived from the first three transformations depicted in Fig. 4 and describes the relation between the 3-D motion and changes of FAP's.

Following the description of the transformations in Section III-B we can set up one 3-D motion equation for each surface point. Instead of the six parameters specifying a rigid body motion, we have a much larger number of unknowns. Rather than estimating the absolute values of the FAP's, we track their evolution over time and estimate relative changes between the previous and the current frame. The analysis–synthesis loop assures that no error accumulation occurs when relative changes are summed up for a long time. In order to obtain a constraint for relative motion between two frames we have to reformulate (2)–(5).

In a first step, a relation between the control point position $\boldsymbol{c}_i'$ of the current frame and the one of the previous frame $\boldsymbol{c}_i$ is established. This relation is given by (2) and (3). To obtain the new control point coordinates, we have to apply the inverse transformations of the previous frame $T^{-1}_{\mathrm{FAP}_0}, \cdots, T^{-1}_{\mathrm{FAP}_{K-1}}$ in descending order and then the transformations of the current frame $T'_{\mathrm{FAP}_0}, \cdots, T'_{\mathrm{FAP}_{K-1}}$ in ascending order

$$\boldsymbol{c}_i' = T'_{\mathrm{FAP}_{K-1}}(\cdots T'_{\mathrm{FAP}_0}(T^{-1}_{\mathrm{FAP}_0}(\cdots(T^{-1}_{\mathrm{FAP}_{K-1}}(\boldsymbol{c}_i))))). \quad (7)$$

This leads to a complicated nonlinear expression which cannot be easily incorporated into a parametric motion estimation framework. However, one may assume that the rotation angles due to changes in the FAP's are relatively small between two successive video frames. The rotation of a control point around an axis with direction $\boldsymbol{\omega}$ can then be substituted by a translation along the tangent

$$T_{\mathrm{FAP}_k}(\boldsymbol{c}_i) = \Delta\boldsymbol{R}_{\mathrm{FAP}_k}(\boldsymbol{c}_i - \boldsymbol{o}_k) + \boldsymbol{o}_k \approx \boldsymbol{c}_i + \boldsymbol{d}_{ik} \cdot \Delta\mathrm{FAP}_k \quad (8)$$

with

$$\boldsymbol{d}_{ik} = s_k(\boldsymbol{\omega} \times (\boldsymbol{c}_i - \boldsymbol{o}_k)). \quad (9)$$

The factor s_k determines the scaling of the corresponding FAP and

$$\Delta\mathrm{FAP}_k = \mathrm{FAP}'_k - \mathrm{FAP}_k \quad (10)$$

is the change of the facial animation parameter k between the two frames. We now have a uniform description for both rotation as well as translation and can estimate both global, as well as local, motion simultaneously. The small error caused by the approximation is compensated after some iterations in the feedback structure shown in Fig. 6. The desired function for the control-point motion results in

$$\boldsymbol{c}_i' = \boldsymbol{c}_i + \sum_k \Delta\mathrm{FAP}_k \hat{\boldsymbol{d}}_{ik} \quad (11)$$

with $\hat{\boldsymbol{d}}_{ik}$ being the 3-D motion-compensated direction vector $\boldsymbol{d}_{ik}$ of the previous frame.

Combination of (5) and (11) leads to the motion equation for an arbitrary surface point as

$$\boldsymbol{x}' = \boldsymbol{x} + \sum_k \boldsymbol{t}_k \Delta\mathrm{FAP}_k = \boldsymbol{x} + \boldsymbol{T} \cdot \boldsymbol{\Delta FAP} \quad (12)$$

where the $\boldsymbol{t}_k$'s are the new direction vectors corresponding to the facial animation parameters which are calculated from $\hat{\boldsymbol{d}}_{ik}$ by applying the linear transforms (4) and (5). $\boldsymbol{T}$ combines all direction vectors in a single matrix of size $3 \times K$ and $\boldsymbol{\Delta FAP}$ is the vector of all FAP changes. The matrix $\boldsymbol{T}$ can be derived from the 3-D model, but has to be set up for each surface point independently. The 3-D motion constraint (12) describes the change of the 3-D point location $\boldsymbol{x}' - \boldsymbol{x}$ as a linear function of FAP changes $\boldsymbol{\Delta FAP}$.

3) Gradient-Based FAP Determination: For the estimation of the facial expression parameters, we use the well-known optical flow constraint equation

$$I_X \cdot u + I_Y \cdot v + I_t = 0 \quad (13)$$

where $[I_X\ I_Y]$ is the gradient of the intensity at point $[X\ Y]$, u and v are the velocity in x- and y-directions, and I_t the intensity gradient in the temporal direction. We can set up (13) for each pixel of a head-and-shoulder scene, but unfortunately, this results in twice as many unknowns u, v as equations. Hence, we need additional constraints to compute a unique solution [22], [24]. Instead of determining the optical flow field by using smoothness constraints and then extracting the motion parameter set from this flow field, we directly estimate the facial animation parameters from (13) by inserting the 3-D motion constraints (12). Our technique is very similar to the one described in [25]. One main difference is that we estimate the motion from synthetic frames and camera images using an analysis-synthesis loop as shown in Fig. 6. This permits a hierarchical framework that can handle larger motion vectors between two successive frames. Another difference between the two approaches is that we use a textured 3-D model to generate new synthetic views of our virtual scene after estimating the motion.

Writing (12) for each component leads to

$$x' = x\left(1 + \frac{1}{x}\boldsymbol{t}_x \cdot \mathbf{\Delta FAP}\right) \tag{14}$$

$$y' = y\left(1 + \frac{1}{y}\boldsymbol{t}_y \cdot \mathbf{\Delta FAP}\right) \tag{15}$$

$$z' = z\left(1 + \frac{1}{z}\boldsymbol{t}_z \cdot \mathbf{\Delta FAP}\right) \tag{16}$$

with $\boldsymbol{t}_x$, $\boldsymbol{t}_y$, and $\boldsymbol{t}_z$ being the row vectors of matrix $\boldsymbol{T}$. Dividing (14) and (15) by (16), incorporating the camera model (6) and using a first order approximation yields

$$u = X' - X \approx -\frac{1}{z}(f_x\boldsymbol{t}_x + (X - X_0)\boldsymbol{t}_z)\mathbf{\Delta FAP} \tag{17}$$

$$v = Y' - Y \approx -\frac{1}{z}(f_y\boldsymbol{t}_y + (Y - Y_0)\boldsymbol{t}_z)\mathbf{\Delta FAP}. \tag{18}$$

These equations serve as the motion constraint in the 2-D image plane. Together with (13), a linear equation for each pixel can be written as

$$\frac{1}{z}(I_X f_x\boldsymbol{t}_x + I_Y f_y\boldsymbol{t}_y + [I_X(X - X_0) + I_Y(Y - Y_0)]\boldsymbol{t}_z)\mathbf{\Delta FAP} = I_t \tag{19}$$

with z being the depth information obtained from the model. This overdetermined system is solved in a least-squares sense with low computational complexity. The size of the system depends directly on the number of FAP's.

Since the system of equations is highly overdetermined, we can discard possible outliers. These outliers are detected by analyzing the partial derivatives of the intensity. Due to the linear approximation of the image intensity, the optical flow constraint (13) is only valid for small displacement vectors. If the estimate of the displacement vector length $\hat{l}$ for the pixel at position $[X\quad Y]$

$$\hat{l}(X, Y) = \sqrt{\frac{I_t^2}{I_X^2 + I_Y^2}} \tag{20}$$

is larger than a threshold, we do not use it for motion estimation.

In order to increase the robustness of the high-dimensional parameter estimation, the range of the parameter space is restricted. Inequality constraints given by matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ specify the allowed range for the FAP's

$$\boldsymbol{A} \cdot \mathbf{FAP} \geq \boldsymbol{a} \tag{21}$$

or restrict the changes in the facial parameters between two successive frames

$$\boldsymbol{B} \cdot \mathbf{\Delta FAP} \geq \boldsymbol{b}. \tag{22}$$

Note that each row of the matrices correspond to one inequality constraint. In our current implementation, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are zero except for a single 1 or -1 in each row leading to a simple restriction of the valid parameter range of the form

$$\begin{aligned} \min_i \leq \mathrm{FAP}_i \leq \max_i \\ \Delta\min_i \leq \Delta\mathrm{FAP}_i \leq \Delta\max_i. \end{aligned} \tag{23}$$

Currently, constraints are used for six facial animation parameters that control the eyelids, the jaw, the lower mid-lip, and the lip corners. The additional terms are incorporated in the optimization of the least-squares problem given by (19) using a *least-squares estimator with inequality constraints* (LSI) [26].

4) Illumination Estimation: The optical flow constraint equation (13) is based on the constant-brightness assumption. However, this assumption is only approximately valid, or, in some cases, not at all. Obviously, if the lighting in the scene changes, we no longer find the same brightness at corresponding object points. But also, if the orientation of the object surface relative to a light source or to the observer changes due to object motion, brightness is in general not constant.

To overcome the constant brightness assumption, we add an illumination component to the scene model that describes the photometric properties for colored light and surfaces. In contrast to the methods proposed in [27] and [28], we incorporate the 3-D information from our head model, which leads to a linear low-complexity algorithm for the estimation of the illumination parameters.

The incident light in the original scene is assumed to consist of ambient light and a directional light source with illumination direction $\boldsymbol{l}$. The surface is modeled by Lambertian reflection, and thus the relation between the video frame intensity I and the corresponding value I_{tex} of the texture map is

$$\begin{aligned} I^R &= I^R_{\mathrm{tex}}(c^R_{\mathrm{amb}} + c^R_{\mathrm{dir}} \cdot \max\{-\boldsymbol{n}\cdot\boldsymbol{l}, 0\}) \\ I^G &= I^G_{\mathrm{tex}}(c^G_{\mathrm{amb}} + c^G_{\mathrm{dir}} \cdot \max\{-\boldsymbol{n}\cdot\boldsymbol{l}, 0\}) \\ I^B &= I^B_{\mathrm{tex}}(c^B_{\mathrm{amb}} + c^B_{\mathrm{dir}} \cdot \max\{-\boldsymbol{n}\cdot\boldsymbol{l}, 0\}) \end{aligned} \tag{24}$$

with c_{amb} and c_{dir} controlling the intensity of ambient and directional light, respectively [23]. The surface normal $\boldsymbol{n}$ is derived from the 3-D head model. The Lambertian model is applied to all three RGB color components separately with a common direction of the incident light. The equations in (24) thus contain eight parameters that characterize the current illumination. By estimating these parameters c^C_{amb}, c^C_{dir} with

$$C \in \{R, G, B\} \tag{25}$$

TABLE I
BITS NEEDED TO ENCODE 3-D MODEL SHAPE FOR DIFFERENTLY QUANTIZED CONTROL POINTS AND THE RESULTING AVERAGE LOSS IN PSNR OF THE DECODED SEQUENCE FROM THE MODEL-BASED CODEC COMPARED TO THE DECODED SEQUENCE WITH NO SHAPE QUANTIZATION (BPC: BITS PER COORDINATE)

bpc	no bits	Δ PSNR
24	16866	-0.0 dB
12	8550	-0.0 dB
10	7164	-0.0 dB
8	5778	-0.0 dB
7	5058	-1.7 dB

we are able to compensate brightness differences of corresponding points in the synthesized frame and the camera frame.

The nonlinear maximum function in (24) complicates the estimation of the illumination parameters. Since the system of equations is overdetermined, we can remove this nonlinearity by excluding object points that are not illuminated by the light source. However, the unknowns in the resulting equations are still nonlinearly coupled. Fortunately, we can break down the estimation process into two linear problems and we employ a least-squares (LS) estimator. First, we determine the illumination direction $\boldsymbol{l}$ and, in a second step, the remaining photometric properties c^C_{amb} and c^C_{dir} are computed.

To obtain the illuminant direction $\boldsymbol{l}$ we divide each component in (24) by its corresponding texture value I^C_{tex} and sum up the three equations. We obtain a system with the four unknowns c_{amb}, $c_{\text{dir}}l_x$, $c_{\text{dir}}l_y$, and $c_{\text{dir}}l_z$

$$\begin{bmatrix}1 & -n_x & -n_y & -n_z\end{bmatrix}\begin{bmatrix}c_{\text{amb}}\\ c_{\text{dir}}\cdot l_x\\ c_{\text{dir}}\cdot l_y\\ c_{\text{dir}}\cdot l_z\end{bmatrix}=\frac{I^R}{I^R_{\text{tex}}}+\frac{I^G}{I^G_{\text{tex}}}+\frac{I^B}{I^B_{\text{tex}}} \quad (26)$$

that is solved in an LS sense. The two values $c_{\text{amb}}=c^R_{\text{amb}}+c^G_{\text{amb}}+c^B_{\text{amb}}$ and $c_{\text{dir}}=c^R_{\text{dir}}+c^G_{\text{dir}}+c^B_{\text{dir}}$ only determine the sum of the desired unknowns c^C_{amb} and c^C_{dir} and are not used in the following. To estimate the single components for each color channel, a second estimation step is necessary. Given the illuminant direction $\boldsymbol{l}$ from the previous estimation, we can set up three independent systems of linear equations for the remaining unknowns c^C_{amb} and c^C_{dir}

$$I^C_{\text{tex}}\cdot\begin{bmatrix}1 & d\end{bmatrix}\begin{bmatrix}c^C_{\text{amb}}\\ c^C_{\text{dir}}\end{bmatrix}=I^C \quad (27)$$

that can again be solved with low complexity in a least-squares sense. We use a nonnegative LS estimator (NNLS) [26] to constrain the coefficients c^C_{amb} and c^C_{dir} to positive values. The reflectivity $d=\max\{-\boldsymbol{n}\cdot\boldsymbol{l},0\}$ is calculated from the previously determined illumination direction and the surface normals from the 3-D model. Note that the images from the video camera are γ-predistorted [29]. This has to be inverted before estimating the photometric properties. The estimated variables are then used for the compensation of the illumination differences between the camera images and the synthetic images using (24) with the appropriate nonlinear mappings to account for γ-predistortion.

5) Encoding of Facial Animation Parameters: In our experiments, 19 FAP's are estimated. These parameters include global head rotation and translation (six parameters), movement of the eyebrows (four parameters), two parameters for eye blinking, and seven parameters for the motion of the mouth and the lips. For the transmission of the FAP's, we predict the current values from the previous frame and quantize the prediction error. An arithmetic coder that is initialized with experimentally determined probabilities is then used to encode the quantized values. Note that the training set for the arithmetic coder is separate from the test set. The parameters for the body motion (four parameters) and the illumination model (eight values) are coded accordingly. The resulting bit stream has to be transmitted as side information.

(a)

(b)

Fig. 7. (a) Frame 120 of the sequence *Peter*. (b) Corresponding model frame.

IV. RATE-CONSTRAINED CODER CONTROL

The coder control employed for the proposed scheme mainly follows the current ITU-T reference model TMN-10 [30], [31] of the H.263 recommendation. As a welcome side effect, we can use an H.263 TMN-10 coder for comparison. In the following, we briefly describe the TMN-10 scheme and explain the extensions to multiframe motion-compensated prediction enabling the incorporation of the model-based coder into H.263.

The problem of optimum bit allocation to the motion vectors and the residual coding in any hybrid video coder is a nonseparable problem requiring a high amount of computation. To circumvent this joint optimization, we split the problem into two parts: motion estimation and mode decision. Motion estimation determines the motion vector and the picture reference parameter to provide the motion-compensated signal. Mode decision determines the macroblock mode considering the rate-distortion

(a)

(b)

Fig. 8. (a) Frame 18 of the sequence *Akiyo*. (b) Corresponding model frame.

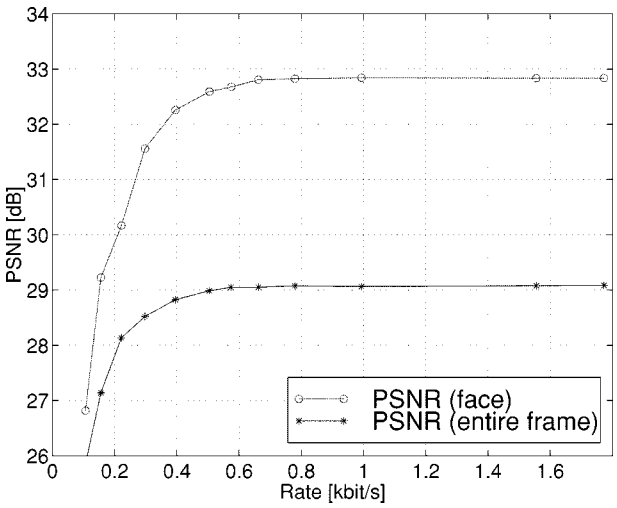


Fig. 9. Rate-distortion plot for the sequence *Peter* using the model-based coder. The two curves show the PSNR measured only in the facial area and over the entire frame.

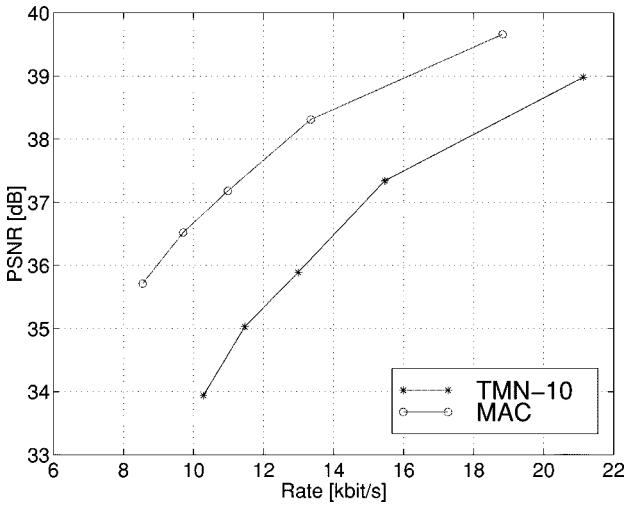


Fig. 10. Rate-distortion plot for the sequence *Peter*.

(a)

(b)

Fig. 11. Frame 120 of the *Peter* sequence coded at the same bit rate using the TMN-10 and the MAC. (a) TMN-10 (33.88 dB PSNR, 1680 bits). (b) MAC (37.34 dB PSNR, 1682 bits).

tradeoff between motion vectors, DCT coefficients, and side information for coder control. Motion estimation and mode decision are conducted for each macroblock given the decisions made for past macroblocks.

Our block-based motion estimation proceeds over both reference frames, i.e., the previous frame and the synthesized model frame. For each block, a Lagrangian cost function is minimized [32], [33] that is given by

$$D_{\mathrm{DFD}}(\boldsymbol{v}, \Delta) + \lambda_{\mathrm{MOTION}} R_{\mathrm{MOTION}}(\boldsymbol{v}, \Delta) \qquad (28)$$

where the distortion D_{DFD} is measured as the sum of the absolute differences (SAD) between the luminance pixels in the original and the block from the reference frame Δ that is displaced by $\boldsymbol{v}$. The term R_{MOTION} is associated with the bit rate

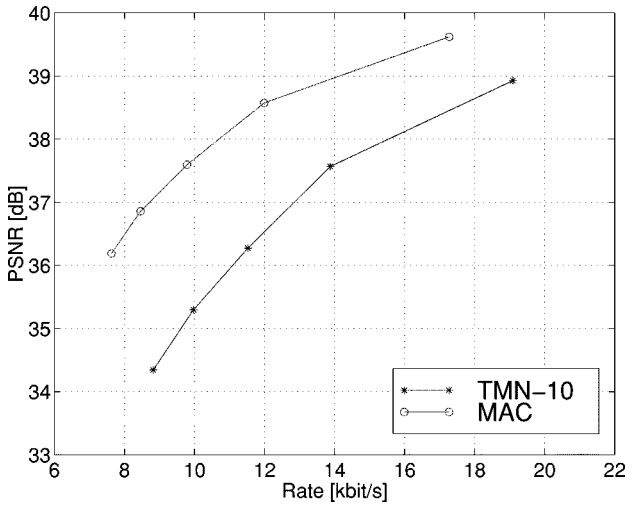


Fig. 12. Rate-distortion plot for the sequence *Eckehard*.

for the motion vector $\boldsymbol{v}$ and the picture reference parameter Δ. The motion vector $\boldsymbol{v}$ is entropy-coded according to the H.263 specification while the picture reference Δ is signaled using one bit. For both frames, the motion search covers a range of ± 16 pixels horizontally and vertically. We have noticed that large displacements are very unlikely to occur when the model frame is referenced leaving further room for optimization.

Given the motion vectors and picture reference parameters, the macroblock modes are chosen. Again, we employ a rate-constrained decision scheme where a Lagrangian cost function is minimized for each macroblock [34]–[36]

$$D_{\mathrm{REC}}(h, \boldsymbol{v}, \Delta, c) + \lambda_{\mathrm{MODE}} R_{\mathrm{REC}}(h, \boldsymbol{v}, \Delta, c). \qquad (29)$$

Here, the distortion after reconstruction D_{REC} measured as the sum of the squared differences (SSD) is weighted against bit rate R_{REC} using the Lagrange multiplier λ_{MODE}. The corresponding rate term is given by the total bit-rate R_{REC} that is needed to transmit and reconstruct a particular macroblock mode, including the macroblock header h, motion information including $\boldsymbol{v}$ and Δ, as well as DCT coefficients c. Based on (29), the coder control determines the best H.263 modes INTER or INTER-4V or INTRA [1] for each macroblock.

Following [37], the Lagrange multiplier for the mode decision is chosen as

$$\lambda_{\mathrm{MODE}} = 0.85 Q^2 \qquad (30)$$

with Q being the DCT quantizer parameter. For the Lagrange multiplier $\lambda_{\mathrm{MOTION}}$, we make an adjustment to the relationship to allow the use of the SAD measure. Experimentally, we have found that an effective method is to measure distortion during motion estimation using SAD rather than the SSD, and to simply adjust the Lagrange multiplier for the lack of the squaring operation in the error computation, as given by $\lambda_{\mathrm{MOTION}} = \sqrt{\lambda_{\mathrm{MODE}}}$.

V. Experimental Results

Experiments are conducted using five natural video sequences. The sequences *Peter*, *Eckehard*, and *Illumination* were recorded in our laboratory and consist of 200, 100, and 150 frames, respectively. All three sequences have CIF resolution (352× 288 pixels) and are encoded at 8.33 frames/s. Additionally, 200 frames of the standard video test sequence *Akiyo* (10 frames/s) and 300 frames of *Claire* (7.5 frames/s) are encoded at CIF resolution.

Since the model-based coder requires the adaptation of the head model to the person in the video sequence, we first

(a)

(b)

Fig. 13. Frame 27 of the *Eckehard* sequence coded at the same bit rate using the TMN-10 and the MAC. (a) TMN-10 (34.4 dB PSNR, 1264 bits). (b) MAC (37.02 dB PSNR, 1170 bits).

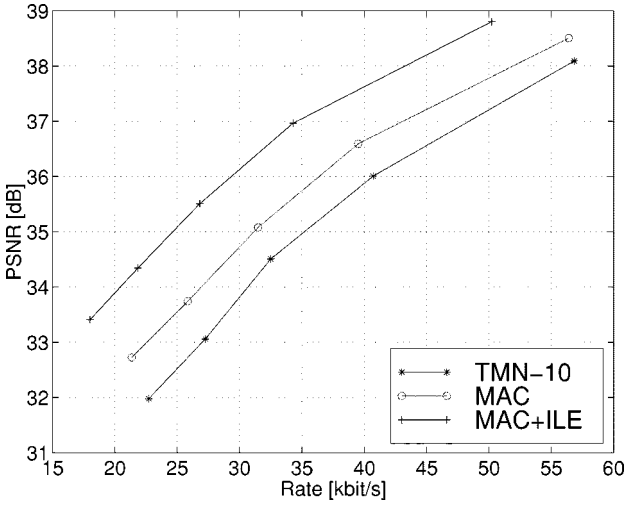


Fig. 14. Rate-distortion plot for the sequence *Illumination* illustrating the achieved improvement when using an illumination estimator (ILE).

(a) (b) (c)

Fig. 15. Frame 99 of the *Illumination* sequence coded at about the same bit rate using the TMN-10, and the MAC without and with illumination compensation. (a) TMN-10 (31.91-dB PSNR, 1960 bits). (b) MAC without illumination compensation (32.45-dB PSNR, 1794 bits). (c) MAC with illumination compensation (34.55-dB PSNR, 1868 bits).

describe the head model initialization used in our experiments. The resulting model is then used to generate model frames with the model-based coder. We briefly present some results for this model-based codec alone. Finally, rate-distortion plots and reconstructed frames are shown for the proposed model-aided coder and compared with the H.263 test model TMN-10.

A. Head Model Initialization

For the shape and texture adaptation of our generic head model toward an individual person, we distinguish between two different cases in our experiments. In the first one, we can utilize explicit shape and texture data from a 3-D scanner to create the head model. In the other case, no shape information of the person in the video sequence is available.

For the three video sequences *Peter*, *Eckehard*, and *Illumination* we have explicit shape and texture information from a 3-D laser scan of the corresponding person. This information is exploited to create the 3-D head model. We map the texture onto the surface and optimize the position of the control points to adapt the spline surface to the measured data. Both shape and texture of the resulting model have to be transmitted initially, unless the decoder has already stored the head model from a previous video-phone session between the two terminals [38]. In all our experiments, we initially position the head model manually, and then rely on automatic tracking by the algorithm described in Section III-C. In a practical system, one would use a technique for automatic face finding and head pose estimation as, e.g., described in [9].

To understand the number of bits needed to transmit an explicit head model we adapt the mesh coding algorithm described in [39] to encode the 231 spline control points of the generic head-model. The topology of the underlying triangle mesh need not be encoded since it is fixed and known at the decoder. To measure the sensitivity of the model-based coding results to the shape quantization we run the model-based codec with differently quantized shape models generating model frames for the sequence *Peter*. The average PSNR in the facial region is computed between the 200 model frames and the original images of the sequence *Peter*. Table I shows the loss in average PSNR for the quantized models compared to the unquantized case. These values show that the shape can be encoded with 5778 bits with no significant loss in quality of the reconstructed sequence. This

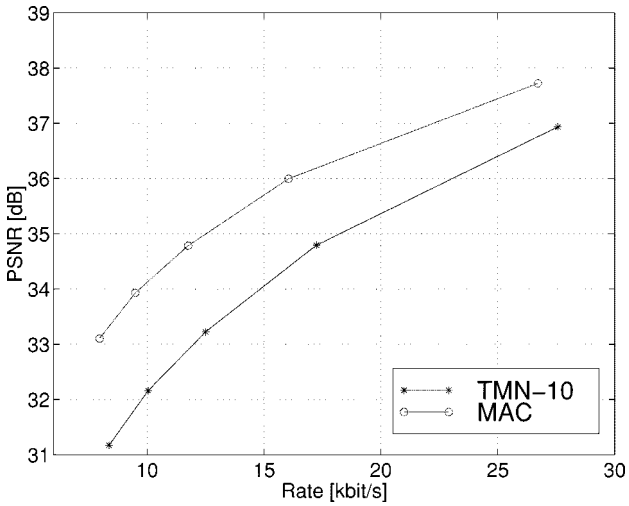
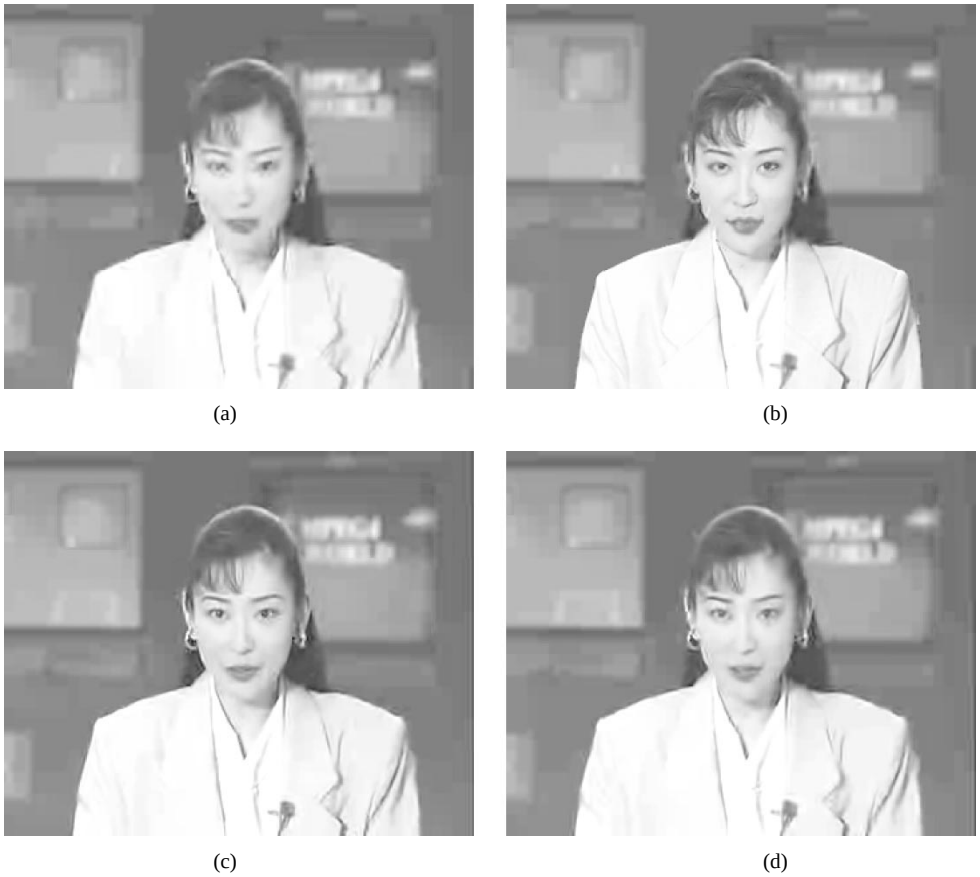


Fig. 16. Rate-distortion plot for the sequence *Akiyo*.

value could be further reduced by predicting the control points from a generic shape model or using more sophisticated coding methods that exploit knowledge about human head shapes [40]. The texture of the explicit head models is coded by H.263 in INTRA mode with a quantizer parameter of 4, requiring 38 480 bits for the sequence *Peter* and 37 928 bits for the sequence *Eckehard*.

For the sequences *Akiyo* and *Claire*, no head shape information from a 3-D scan is available. Hence, the 3-D model is generated using a generic head model. To obtain a better fit of the chin contour, 16 of the 231 control points are modified manually. This solution has been chosen due to its simplicity. Again, sophisticated automatic techniques for initial model adaptation can be found in the literature (e.g., [9], [25]) and would be used in a practical system. Nevertheless, even with our simple and inaccurate approach to initial model adaptation, we are able to demonstrate the practicability of a model-aided codec that does not require a 3-D scan.

Once the shape of the generic head model is adapted, the texture for the *Akiyo* and *Claire* model is extracted from the first INTRA frame that is encoded with a quantizer parameter of 4 in the face and body region. 45 472 bits are necessary to encode the first INTRA frame for *Claire*, the corresponding frame for *Akiyo* is coded with 51 280 bits. Those parts of the texture not



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

Fig. 17. Frame 150 of the *Akiyo* sequence. (a) TMN-10 (31.08-dB PSNR, 720 bits). (b) MAC (33.19-dB PSNR, 725 bits). (c) TMN-10 (34.7-dB PSNR, 1304 bits). (d) TMN-10 (33.12-dB PSNR, 912 bits).

visible in the first frame are extrapolated from the visible parts. In our experiments the texture is neither changed nor updated after the first frame.

B. Model-Based Coding

Using the individualized 3-D head model, the FAP's and illumination parameters are estimated and encoded with the model-based coder described in Section III. At coder and decoder side, the model is animated according to the quantized FAP's and a synthetic sequence of model frames is rendered. Fig. 7 shows one original frame of the sequence *Peter* and its synthetic counterpart. Since the 3-D model does not provide reliable information about the hair and the inside of the mouth, model failures lead to errors in the synthesized frame that have to be compensated by the block-based coder. A similar result is illustrated in Fig. 8 showing an original and the corresponding model frame for the *Akiyo* sequence.

The quality of the synthetic model-based prediction is illustrated in Fig. 9 where the PSNR, measured only in the facial area, is plotted over the bit rate needed for encoding the sequence *Peter*. Note that model failures lead to a saturation of the curve at 32.8 dB. The bit rate necessary to reach that point is below 1 kb/s. For comparison, the corresponding plot with PSNR measured over the entire frame is also depicted in Fig. 9. The latter curve is mainly characterized by the erroneous background and the hair region, which are not expected to be utilized for prediction in the model-aided coder. Note that the PSNR in the following is always measured over the entire frame.

C. Model-Aided Coding

For comparing the proposed model-aided coder with TMN-10, the state-of-the-art test model of the H.263 standard, rate-distortion curves are measured by varying the DCT quantizer parameter over values 10, 15, 20, 25, and 31. Decodable bit-streams are generated that produce the same PSNR values at the encoder and decoder. In our simulations, the data for the first INTRA frame and the initial 3-D model are excluded from the results. In addition, the results for the first 30 frames are excluded. Thus, we compare the interframe coding performance of both codecs without the start-up phase at the beginning of the sequence.

In the following, we show rate-distortion curves for the proposed model-aided coder in comparison to the H.263 test model, TMN-10. Additionally, subjective results by means of comparing reconstructed frames are presented, since we have found that the PSNR measure allows only limited conclusions about the comparisons.[1]

The following abbreviations are used for the two codecs compared.

1) *TMN-10:* The result produced by the H.263 test model, TMN-10, using Annexes D, F, I, J, and T.

[1]The decoded frames can also be found [Online]. Available: www.lnt.de/~eisert/mac.html.

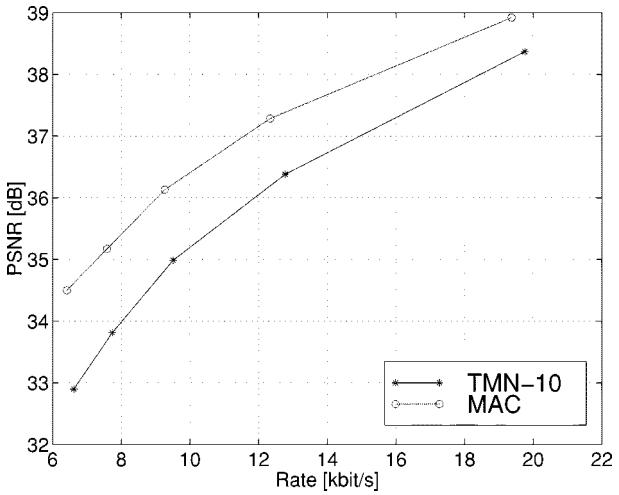


Fig. 18. Rate-distortion plot for the sequence *Claire*.

2) *MAC:* Model-aided coder—H.263 extended by model-based prediction with Annexes D, F, I, J, and T enabled as well.

Fig. 10 shows the rate-distortion curves obtained for sequence *Peter*. Significant gains in coding efficiency are achieved compared to TMN-10. Bit-rate savings of about 35% at equal average PSNR are visible at the low bit-rate end. This corresponds to a gain of about 2.8 dB in terms of average PSNR. From the same sequence, frame 120 is depicted in Fig. 11. The upper picture is decoded and reconstructed from the TMN-10 decoder, while the lower picture corresponds to the model-aided coder. Both frames require about the same number of bits and are taken from sequences that are approximately encoded at the same bit rate using a quantizer parameter of 31 for the TMN-10 and a value of 25 for the model-aided coder.

Similar gains in coding efficiency are obtained for the sequence *Eckehard* as can be seen in Fig. 12. Two decoded frames at equal bit rate are depicted in Fig. 13. The upper image corresponds to frame 27 and is decoded and reconstructed from the TMN-10 decoder, while the lower one is generated from the model-aided coder.

The effectiveness of the illumination estimation is illustrated in Fig. 14 for the sequence *Illumination*. During the acquisition of this sequence, one light source was moved to alter the illumination conditions. Two experiments are performed. For the first one, only the FAP's are estimated to create a model frame. For the second experiment, we additionally estimate the illumination parameters as described in Section III-C-4 and generate motion- and illumination-compensated model frames. As shown in Fig. 14, the gain in PSNR for the model-aided coder compared to the TMN-10 is about 1 dB if no illumination compensation is performed. However, an additional gain of about 1.5 dB is achieved when exploiting illumination information. Corresponding decoded frames for all three cases are shown in Fig. 15. The additional computation for the illumination estimation is negligible compared to the other codec components since only linear systems of equations with up to four unknowns have to be solved. For the other video sequences, which show a constant illumination, rather marginal gains of about 0.2 dB are achieved when estimating photometric properties from the scene.

(a)

(b)

Fig. 19. Frame 60 of the *Claire* sequence using the TMN-10 and the MAC. (a) TMN-10 (33.21-dB PSNR, 752 bits). (b) MAC (35.05-dB PSNR, 761 bits).

TABLE II
PERCENTAGE OF MACROBLOCKS SELECTED FROM THE MODEL FRAME FOR DIFFERENT QUANTIZER VALUES AND VIDEO SEQUENCES

quant	31	25	20	15	10
Peter	17.3%	17.3%	17.4%	16.4%	14.3%
Eckehard	12.7%	13.2%	13.3%	13.1%	12.2%
Illumination	29.3%	28.8%	27.1%	25.1%	21.8%
Akiyo	7.6%	7.8%	7.7%	7.5%	6.7%
Claire	6.5%	5.9%	5.5%	5.4%	5.2%

Up to now, the experiments are performed with video sequences where explicit head shape information is available from a 3-D scanner. This is not the case for the next two video sequences. Fig. 16 shows results for the sequence *Akiyo*. For this sequence, the bit-rate savings are still about 35% at the low bit rate end. The quality of the reconstructed frames is shown in Fig. 17. The upper-right image shows frame 150 encoded with the model-aided coder, while the upper left image corresponds to the TMN-10 coder at the same bit rate. At the lower right of Fig. 17, a frame from the TMN-10 coder is shown that has the same PSNR as the upper model-aided frame. Even though the PSNR is the same, the subjective quality of the reconstructed frame from the model-aided coder is clearly superior since facial features are reproduced more accurately and with less artifacts.

TABLE III
MODE PERCENTAGE FOR THE FOUR MODES "UNCODED," "INTER," "INTER4V," AND "INTRA" FOR THE MODEL-AIDED CODER. TWO DIFFERENT QUANTIZER PARAMETERS (31 AND 15) ARE CHOSEN FOR THE SEQUENCES PETER (PE), ECKHARD (EK), ILLUMINATION (IL), AKIYO (AK), AND CLAIRE (CL)

	PE 31	PE 15	EK 31	EK 15	IL 31	IL 15	AK 31	AK 15	CL 31	CL 15
UNCODED	92.2%	84.3%	95.0%	89.0%	77.4%	59.6%	95.9%	89.4%	93.9%	86.6%
INTER	7.0%	14.3%	4.8%	10.1%	9.6%	21.3%	3.9%	8.3%	5.6%	11.0%
INTER4V	0.1%	1.0%	0.1%	0.7%	0.7%	2.5%	0.3%	2.3%	0.5%	2.3%
INTRA	0.6%	0.4%	0.1%	0.2%	12.3%	16.6%	0.0%	0.0%	0.0%	0.0%

The difference is even more striking when viewing motion sequences. Finally, the lower left image is encoded with TMN-10 to yield the same subjective quality as the model-aided coder; TMN-10 requires about twice as many bits.

The corresponding rate-distortion plot and the decoded frames for the sequence *Claire* are shown in Figs. 18 and 19. For this sequence the bit-rate savings are about 27% at the low bit-rate end. The slightly smaller gain is due to the small face area of *Claire* and the limited motion in the sequence.

The gain in PSNR and the reduction in average bit rate depend on the number of macroblocks that are selected from the model frame to predict the current video frame. These blocks are motion-compensated by the model-based coder saving bits for the motion vector and for the residual coding. Table II shows the percentage of macroblocks that are selected from the model frame. The percentage of the different coding modes for the model-aided coder is illustrated in Table III.

Fig. 20 shows the model-based prediction of frame 18 that was already shown in Fig. 8. As can be seen in the enlargement of the region around the mouth, a model failure occurs that causes the black bar inside the mouth. The rate-constrained coder control handles such model failures automatically, as illustrated in Fig. 21. Fig. 21 shows all macroblocks that are predicted from the model frame, while the macroblocks predicted from the previous frame are grey. The mouth is not predicted from the model frame thus avoiding the prediction error coding of the black bar in Fig. 20.

Finally, we address the computational complexity of the model-aided coder. The CIF sequence *Claire* is encoded on a 175-MHz O_2 SGI workstation with R10000 processor and the average time for processing a frame is measured. Note that the implementation is not a real-time implementation and many optimizations are possible without changing the underlying algorithms. The complete processing time for the encoding of one frame is 38.0 s. 13.8 s of that time is spent for the block-based coder with motion compensated prediction from two frames. For comparison, the encoding using our software implementation of a H.263 coder with prediction from one frame takes 13.0 s. To encode and decode a model frame, the model-based part of the coder needs 24.2 s, including rendering. Note that the graphics hardware of the SGI workstation is not utilized in our implementation. The most demanding component of the model-based coder is the FAP estimator whose execution time is mainly determined by the setup of the system of equation (19) and its solution using the LSI estimator. For this task, 6.7 s are necessary on average for all iterations in the analysis–synthesis loop. The ILE described in Section III-C-4 needs only 0.2 s, and does not alter the total processing time noticeably.

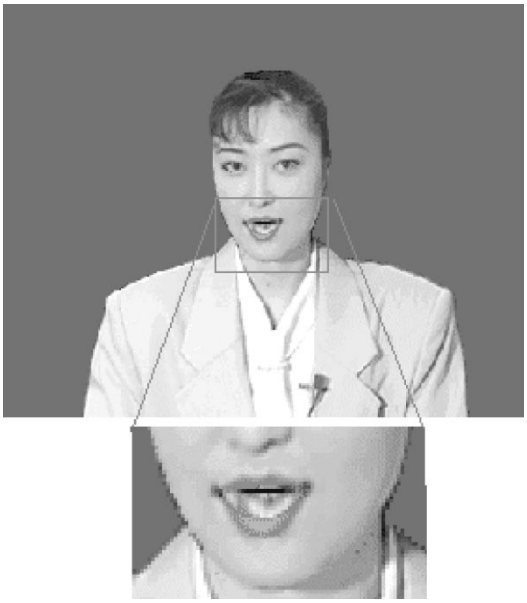


Fig. 20. Model frame 18 of the sequence *Akiyo* with enlargement of the image region around the mouth.

Fig. 21. Frame 18 of the sequence *Akiyo*. Macroblocks shown have been selected for motion compensated prediction using the model frame. The grey parts of the image are predicted from the previous frame, or they are coded in INTRA mode.

VI. CONCLUSION

We presented a new approach to incorporate facial animation into motion-compensated video coding of head-and-shoulder sequences. This is achieved by combining a model-based coder

with a block-based hybrid coder such as H.263 in a rate-distortion-efficient framework. The model-based coder estimates the facial expressions and the 3-D motion of a person using a 3-D head model. Since only a few parameters are encoded and transmitted, very low bit rates, typically less than 1 kb/s, are obtained if the 3-D models can describe the current video frame. Standard block-based hybrid coders are not restricted to a special scene content, but they are much less efficient.

The advantages of both approaches are combined in a new framework by employing the synthesized frame from the model-based coder as a second reference frame for rate-constrained block-based motion-compensated prediction in addition to the previously reconstructed reference frame. For each block, the video coder decides which of the two frames to select for motion compensation using a Lagrangian cost function. This multiframe prediction and the combined encoding of the different modes provides increased robustness to model failures in the model-based coder and ensures the generality of the approach.

In our experimental results, we showed that bit-rate savings around 35% are achieved at equal average PSNR in comparison to TMN-10, the state-of-the-art test model of the H.263 video-compression standard, for head-and-shoulder sequences. At equal PSNR, the subjective quality of the reconstructed frame from the model-aided coder is clearly superior since facial features are reproduced more accurately and with less artifacts.

These results show that waveform-coding and 3-D model-based coding are not competing alternatives but should support and complement each other. Both can be elegantly combined in a rate-distortion framework, such that the generality of waveform coding and the efficiency of 3-D models are available where needed.

REFERENCES

[1] *Video coding for low bitrate communication*, ITU-T Recommendation H.263 Version 2 (H.263+), Jan. 1998.

[2] *Generic Coding of Audio-Visual Objects: (MPEG-4 video)*, Final Draft Int. Standard, Document N2502, 1999.

[3] W. J. Welsh, S. Searsby, and J. B. Waite, "Model-based image coding," *British Telecom Technol. J.*, vol. 8, no. 3, pp. 94–106, July 1990.

[4] D. E. Pearson, "Developments in model-based video coding," *Proc. IEEE*, vol. 83, pp. 892–906, June 1995.

[5] P. Eisert and B. Girod, "Analyzing facial expressions for virtual conferencing," *IEEE Comput. Graph. Applicat.*, vol. 18, pp. 70–78, Sept. 1998.

[6] E. Reusens, R. Castagno, C. le Buhan, L. Piron, T. Ebrahimi, and M. Kunt, "Dynamic video coding—An overview," in *Proc. Int. Conf. Image Processing*, vol. 2, Lausanne, Switzerland, Sept. 1996, pp. 377–380.

[7] M. F. Chowdhury, A. F. Clark, A. C. Downton, E. Morimatsu, and D. E. Pearson, "A switched model-based coder for video signals," *IEEE Trans. Circuits Syst. Video Technol.* , vol. 4, pp. 216–227, June 1994.

[8] H. G. Musmann, "A layered coding system for very low bit rate video coding," *Signal Processing: Image Commun.*, vol. 7, no. 4–6, pp. 267–278, Nov. 1995.

[9] M. Kampmann and J. Ostermann, "Automatic adaptation of a face model in a layered coder with an object-based analysis-synthesis layer and a knowledge-based layer," *Signal Processing: Image Communication*, vol. 9, no. 3, pp. 201–220, Mar. 1997.

[10] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 70–84, Feb. 1999.

[11] T. Wiegand, E. Steinbach, A. Stensrud, and B. Girod, "Multiple reference picture coding using polynomial motion models," in *Proc. SPIE Conf. Visual Communications and Image Processing*, San Jose, CA, Feb. 1998, pp. 134–145.

[12] E. Steinbach, T. Wiegand, and B. Girod, "Using multiple global motion models for improved block-based video coding," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999.

[13] H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, 1963.

[14] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[15] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.

[16] M. Rydfalk, "CANDIDE: A parameterized face," Ph.D. dissertation, Linköping University, Linköping , Sweden, 1978.

[17] G. Greiner and H. P. Seidel, "Splines in computer graphics: Polar forms and triangular B-spline surfaces," *Eurographics*, 1993.

[18] M. Hoch, G. Fleischmann, and B. Girod, "Modeling and animation of facial expressions based on B-splines," *Vis. Comput.*, vol. 11, pp. 87–95, 1994.

[19] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics, Principles and Practice*, 2nd ed. Reading, MA: Addison-Wesley, 1990.

[20] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3-D machine vision metrology using off-the-shelf cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, pp. 323–344, Aug. 1987.

[21] P. Eisert and B. Girod, "Model-based estimation of facial expression parameters from image sequences," in *Proc. Int. Conf. Image Processing*, vol. 2, Santa Barbara, Oct. 1997, pp. 418–421.

[22] H. Li, P. Roivainen, and R. Forchheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 545–555, June 1993.

[23] P. Eisert and B. Girod, "Model-based coding of facial image sequences at varying illumination conditions," in *Proc. 10th Image and Multidimensional Digital Signal Processing Workshop IMDSP '98*, Alpbach, Austria, July 1998, pp. 119–122.

[24] J. Ostermann, "Object-based analysis-synthesis coding (OBASC) based on the source model of moving flexible 3-D-objects," *IEEE Trans. Image Processing*, vol. 3, pp. 705–711, Sept. 1994.

[25] D. DeCarlo and D. Metaxas, "The integration of optical flow and deformable models with applications to human face shape and motion estimation," *Comput. Vis. Pattern Recognit.*, pp. 231–238, 1996.

[26] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[27] J. Stauder, "Estimation of point light source parameters for object-based coding," *Signal Processing: Image Commun.*, vol. 7, no. 4/6, pp. 355–379, Nov. 1995.

[28] G. Bozdagi, A. M. Tekalp, and L. Onural, "3-D motion estimation and wireframe adaption including photometric effects for model-based coding of facial image sequences," *IEEE Trans. Circuits Systems Video Technol.*, vol. 4, pp. 246–256, June 1994.

[29] C. A. Poynton, "Gamma and its disguises: The nonlinear mappings of intensity in perception, crts, film and video," *SMPTE J.*, pp. 1099–1108, Dec. 1993.

[30] *Video Codec Test Model, Near Term, Version 10 (TMN-10), Draft 1*, Download via anonymous ftp to: standard.pictel.com/video-site/9804_Tam/q15d65d1.doc, Apr. 1998.

[31] *An Improved H.263-Codec using Rate-Distortion Optimization*, Available FTP: standard.pictel.com/video-site/9804_Tam/q15d13.doc, Apr. 1998.

[32] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, May 1991, pp. 2713–2716.

[33] B. Girod, "Rate-constrained motion estimation," in *Proc. SPIE Conf. Visual Communications and Image Processing*, Chicago, IL, Sept. 1994, pp. 1026–1034.

[34] T. Wiegand, M. Lightstone, T. G. Campbell, and S. K. Mitra, "Efficient mode selection for block-based motion compensated video coding," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, Oct. 1995, pp. 559–562.

[35] G. M. Schuster and A. K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263," in *Proc. SPIE Conf. Visual Communications and Image Processing*, Orlando, FL, Mar. 1996, pp. 784–795.
[36] 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, pp. 182–190, Apr. 1996.
[37] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.
[38] B. Girod, "Image sequence coding using 3-D scene models," in *Proc. Visual Computation and Image Processing (VCIP)*, vol. 3, Sept. 1994, pp. 1576–1591.
[39] M. Magnor and B. Girod, "Fully embedded coding of triangle meshes," in *Proc. Vision, Modeling, and Visualization (VMV'99)*, Erlangen, Germany, Nov 1999, pp. 253–259.
[40] T. Vetter and V. Blanz, "Estimating colored 3-D face models from single images: An example based approach," in *Proc. Euro. Conf. Computer Vision (ECCV)*, 1998.

Peter Eisert received the Dipl.-Ing. degree in electrical engineering from the University of Karlsruhe, Germany, in 1995.

He then joined the Image Communication Group at the Telecommunications Institute, University of Erlangen-Nuremberg, Germany. As a member of the Center of Excellence "3-D Image Analysis and Synthesis," he is currently working on his Ph.D. thesis. His research interests include model-based video coding, 3-D object reconstruction, image communication and computer vision.

Thomas Wiegand (S'92–M'93) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Germany, in 1995. He is currently working toward the Dr.-Ing. degree at the University of Erlangen-Nuremberg, Germany.

From 1993 to 1994, he was a Visiting Researcher with Kobe University, Japan. In 1995, he was a Visiting Scholar at University of California at Santa Barbara. From 1997 to 1998, he was a Visiting Researcher at Stanford University, Stanford, CA. His research interests are in the field of information and estimation theory, with focus on digital video compression and transmission. He is an active contributor to ITU-T/SG16/Q15.

Bernd Girod (S'83–M'89–SM'97–F'98) received the Engineering Doctorate from University of Hannover, Germany, and the M.S. degree from Georgia Institute of Technology, Atlanta, GA

He is Chaired Professor of Telecommunications in the Electrical Engineering Department, University of Erlangen-Nuremberg, Germany. Since 1996, he has served as Director of the Center of Excellence "3-D Image Analysis and Synthesis" in Erlangen. He has held prior visiting and regular faculty positions with MIT, Georgia Tech, and Stanford. He has been involved with several start-up ventures, among them PictureTel, Polycom, Vivo Software, 8x8, and RealNetworks. His research interests are in the areas of image communication, 3-D image analysis and synthesis, and multimedia systems.

Prof. Girod was recently elected Fellow of the IEEE for his contributions to the theory and practice of video communications.