MPEG-4 Facial Animation in Video Analysis and Synthesis

Peter Eisert
Image Processing Department
Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institute
Einsteinufer 37, D-10587 Berlin, Germany
E-mail: eisert@hhi.de
URL: http://bs.hhi.de/~eisert

Abstract—
MPEG-4 supports the definition, encoding, transmission, and animation of 3-D head and body models. These features can be used for a variety of different applications ranging from low bit-rate video coding to character and avatar animation. In this paper, an entire system for the analysis of facial expressions from image sequences and their synthesis is presented. New methods for the estimation of MPEG-4 facial animation parameters as well as scene illumination are proposed. Experiments for different applications demonstrate the potential of using facial animation techniques in video analysis and synthesis. A model-based codec is presented that is able to encode head-and-shoulder video sequences at bit-rates of about 1 kbit/s. Besides the low bit-rate, many enhancements and scene modifications can be easily applied, like scene lighting changes or cloning of expressions for character animation. But also for the encoding of arbitrary sequences, 3-D knowledge can help to increase the coding efficiency. With our model-aided codec, bit-rate reductions of up to 45 % at the same quality can be achieved in comparison to standard hybrid video codecs.

Keywords— **MPEG-4, Facial Animation, Video Compression, Model-Based Coding, Model-Aided Coding**

I. Introduction

THE human face is one of the most interesting object for our visual perception. Incessantly in our lives, we are confronted with faces and even from a short glance, we are able to interpret them. Besides the astonishing capability to recognize individuals from a large number of faces, an enormous amount of information about people can be gathered when looking at them even briefly. Mood or attitude of a person can be determined from very subtle changes in the facial mimic. Faces attract our interest and provide familiarity which is often exploited in illustrations, image sequences, or other visual representations. Movies or video sequences would be less interesting without any "emotional" content and the focus of attention is often attracted to the faces in a scene.

On the technical side, researchers have worked for a long time on creating realistic synthetic facial animation. This task is complicated by the extreme human sensitivity towards small errors in the facial mimic. Even small deviations from our expectation result in artificially looking animations. However, in the last years, a lot of progress has been made in facial modeling and rendering, and today highly realistic facial animation can be found in many applications. Off-line rendering of virtual characters in movie productions, for example, is familiar to us, but the increasing computational and graphical power enables the real-time usage of facial animation techniques also on small devices like personal digital assistants (PDAs) or smart phones. The applicability of these techniques is not restricted to the creation of synthetic graphics content but they can also help to improve the efficiency and quality in communication applications. Handheld devices are often connected to the Internet over low-bandwidth wireless channels. In order to enable video-phone applications or streaming of video clips also for these low bit-rate scenarios, high compression ratios of the raw data are required.

The problem of efficiently compressing video data has been approached by several video coding standards. Hybrid video codecs as described by H.261/263/264 or MPEG-1/-2/-4 use block based motion compensated prediction in combination with residual coding to achieve compression ratios of several 100:1 at acceptable visual quality. If much higher coding gains are required more sophisticated scene models can be used which exploit a-priori knowledge that is available to both encoder and decoder and that need not be transmitted. Model-based codecs [11], [32], [24], e.g., use three-dimensional computer models to describe the objects in the scene. This representation is transmitted only once and all following scene changes are described by motion, deformation, or lighting parameters. Since only a few parameters have to be encoded for each frame of the image sequence, extremely low bit-rates can be achieved. In order to restrict the number of 3-D models for transmission, model-based coding is especially suited for applications like video-phones, video-conferencing, and streaming of video content like news casts. In these scenarios, usually one or more people are visible in the foreground while the background remains rather static or is of limited interest. The scene can be represented by 3-D head-and-shoulder models that are animated in order to show motion and facial expressions. Both, head model animation and 3-D background modeling is supported in MPEG-4 [19] enabling standard-compliant model-based coding.

Besides the low bit-rate of model-based systems that can be exploited for image communication over low band-width wireless channels (e.g., GPRS, UMTS), the semantic information about the objects simplifies the modification of scene content and enables new applications. In immersive

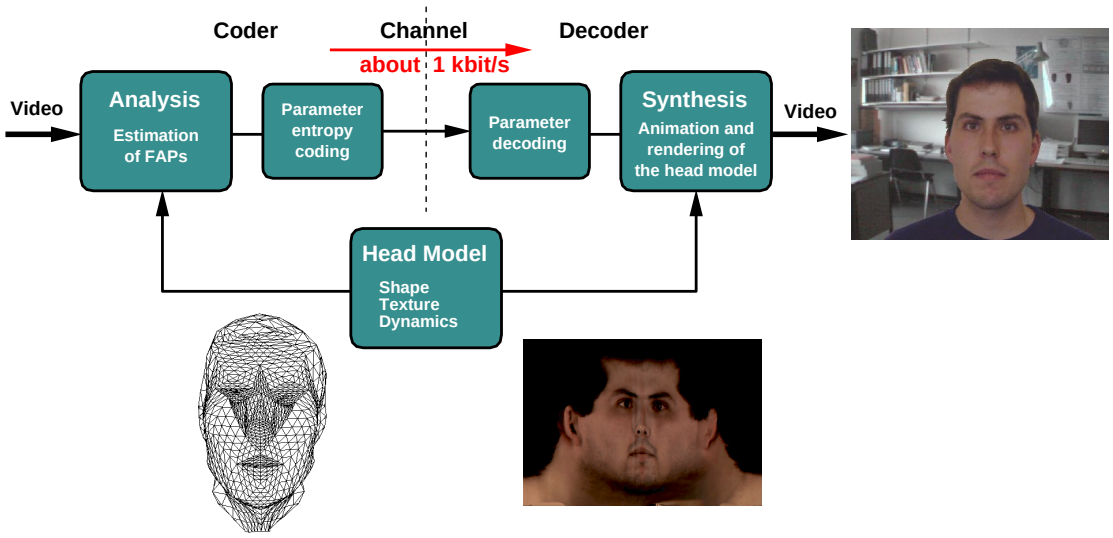

Fig. 2. Model-based Video Codec.

Fig. 1. Immersive teleconferencing application.

video-conferencing [17], multiple participants who are located at different places can be seated at a joint virtual table. Due to the 3-D representation of the objects, pose modification for correct seating positions can easily be accomplished as well as view-point corrections according to the user's motion. By replacing the 3-D model of one person by a different one, other people can be animated with the expressions of an actor. This can be exploited in film productions for marker- and sensor-less motion capture and to synthesize new sequences of artificial creatures or an already passed away person. Similarly, avatars can be driven to create user-friendly man-machine-interfaces where a human-like character interacts with the user. Analyzing the user with a web cam gives the computer also feedback about the user's emotions and intentions [25]. Other cues in the face can assist the computer-aided diagnosis and treatment of patients in medical applications. For example, asymmetry in facial expressions caused by facial palsy can be measured three-dimensionally [12] or craniofacial syndromes can be detected by the 3-D analysis of facial feature positions [14]. These examples indicate the wide variety of applications for model-based facial analysis and synthesis techniques.

In this paper, a new approach for facial expression analysis and synthesis is presented. First in Section II, the concept of model-based coding is introduced. This coding scheme requires the synthesis of 3-D model descriptions which is described in the context of MPEG-4 in Section II-A. In Section II-B, the linear and robust estimation of MPEG-4 facial animation parameters is addressed. Since lighting changes can severely affect the performance of such motion tracking techniques, photometric properties are also estimated from the image sequence as shown in Section II-C. Then, experiments for different applications are presented in Section II-D like, model-based coding, lighting enhancement, expression cloning, and view morphing. Finally, the concept of model-aided coding is described in Section III together with experiments illustrating the improvements in coding efficiency compared to state-of-the-art hybrid video codecs.

II. Model-based Coding

In model-based coding [11], [32], [24], three-dimensional computer models define the appearance of all objects in the scene and only high level motion, deformation, and lighting information is streamed to represent the dynamic changes in the sequence. For the particular case of head-and-shoulder video sequences, the structure of a model-based codec is depicted in Fig. 2. A camera captures frames from a person participating, e.g., in a virtual conference. From the first frames, a 3-D head model is created specifying shape and texture of the person. This model is encoded and transmitted only once if it has not already been stored at the decoder in a previous session. The encoder then analyzes the video sequence and estimates 3-D motion and facial expressions using the 3-D head model information. The expressions are represented by a set of facial animation

parameters (FAPs) which are streamed over the network. At the decoder, the 3-D head model is deformed according to the FAPs and new frames are synthesized using computer graphics techniques. Since only a few parameters have to be transmitted for each frame of the sequence, bit-rates of about 1 kbit/s can be achieved [6], [4].

A. Facial Expression Synthesis and Animation

For the synthesis of head-and-shoulder video sequences, a 3-D head model must be defined which is deformed over time to show facial expressions [28], [23], [26]. Both head model specification and streaming of facial animation parameters are addressed in the MPEG-4 standard [19]. This allows building a standard compliant model-based codec that can stream head-and-shoulder video sequences at bit-rates of a few kbit/s. The decoding complexity for such a system is rather limited and real time rendering can be realized on current PDAs at interactive frame rates.

The MPEG-4 standard provides three different ways of defining 3-D head models:

1. A default model can be used that is already available at the decoder. No geometry or texture information has to be streamed initially. This mode can be used for low bit-rate animation of avatars or virtual agents, but no control to the appearance of the model can be applied.
2. In order to adapt the model to the appearance of the person in the video, MPEG-4 defines 84 *facial definition points* (FDPs). These points represent the position of distinct facial features and enable the decoder to deform the facial mesh. The 3-D location of these points have to be transmitted to the decoder for a rough shape adaptation of the available head model. In addition, a texture map can be specified for increased realism of the synthetic model.
3. Both aforementioned approaches have the disadvantage that the encoder is not aware of the exact geometry of the model at the decoder. This is avoided if the entire mesh is transmitted to the decoder. In MPEG-4, triangle meshes and texture maps can be specified using the VRML-like scene graph description BIFS (binary format for scenes). With these elements, arbitrarily shaped head models can be defined which are animated using a *face animation table* (FAT). This table is included in BIFS and defines for each *facial animation parameter* the magnitude and direction of all 3-D vertex movements caused by changes of the corresponding parameter. Given the static textured mesh representation and the FAT, the head model can be controlled at encoder and decoder identically.

In the following, only the third case is considered and it is assumed that the same textured head model is available at encoder and decoder. The shape of a generic 3-D model [6] is adapted to the person in the first frame of the video sequence. Texture is projected onto the model and extrapolated in non-visible areas. Fig. 3 shows a wireframe and the corresponding textured version of the already adjusted head model. Additionally, a FAT is constructed that defines the mapping from FAPs to surface deformations in order to allow local motion caused by facial expressions. Since the topology of the generic model remains constant, the FAT need not be changed for different people. In principle, this table also allows to use different parameter sets and more efficient orthogonal facial expression representations like those proposed in [31] and [16].

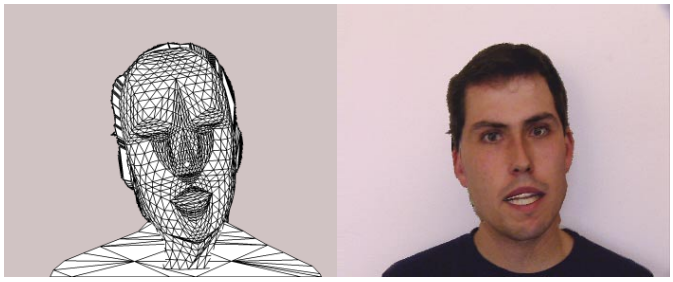

Fig. 3. Wireframe representation and textured 3-D head model.

The dynamic changes in a synthesized video sequence are completely described by the set of *facial animation parameters*. Each parameter defines an elementary motion field that affects a local part in the face like the motion of lip corners, eye movement, or head rotation. The final facial expression is generated by superposing all these action units. MPEG-4 defines 66 different parameters to control facial motion. Not all of these parameters need to be used but subsets can be selected in order to trade complexity and bit-rate with motion accuracy and realism. For a streaming scenario, the parameters are predicted from the previous frame, quantized, and encoded with an arithmetic coder. The alternative method of using DCT coding for the FAPs is not considered here, since additional delay is introduced in that case.

B. Facial Expression Analysis

The most challenging part of facial expression analysis is the estimation of 3-D facial motion and deformation from two-dimensional images. Due to the loss of one dimension caused by the projection of the real world onto the image plane, this task can only be solved by exploiting additional knowledge of the objects in the scene. In particular, the way the objects move can often be restricted to a low number of degrees of freedom that can be described by a limited set of parameters. In this section, a new 3-D model-based method for the estimation of facial expressions is presented that makes use of an explicit parameterized 3-D human head model describing shape, color, and motion constraints of an individual person [4]. This model information is jointly exploited with spatial and temporal intensity gradients of the images. Thus, the entire area of the image showing the object of interest is used instead of dealing with discrete feature points, resulting in a robust and highly accurate system. A linear and computationally efficient algorithm is derived for different scenarios. The scheme is embedded in a hierarchical analysis-synthesis framework to avoid error accumulation in the long-term estimation.

B.1 Optical-flow based Analysis

In contrast to feature-based methods, gradient-based algorithms utilize the optical flow constraint equation

$$\frac{\partial I(X,Y)}{\partial X}d_x + \frac{\partial I(X,Y)}{\partial Y}d_y = I(X,Y) - I'(X,Y), \quad (1)$$

where $\frac{\partial I}{\partial X}$ and $\frac{\partial I}{\partial Y}$ are the spatial derivatives of the image intensity at pixel position $[X\ Y]$. $I' - I$ denotes the temporal change of the intensity between two time instants $\Delta t = t' - t$ corresponding to two successive frames in an image sequence. This equation, obtained by Taylor series expansion up to first order of the image intensity, can be set up anywhere in the image. It relates the unknown 2-D motion displacement $\mathbf{d} = [\mathbf{d_x},\ \mathbf{d_y}]$ with the spatial and temporal derivatives of the images.

The solution of this problem is under-determined since each equation has two new unknowns for the displacement coordinates. For the determination of the optical flow or motion field, additional constraints are required. Instead of using heuristical smoothness constraints, explicit knowledge about the shape and motion characteristics of the object in exploited. Any 2-D motion model can be used as an additional motion constraint in order to reduce the number of unknowns to the number of motion parameters of the corresponding model. In that case, it is assumed that the motion model is valid for the complete object. An overdetermined system of equations is obtained that can be solved robustly for the unknown motion and deformation parameters in a least-squares sense.

In the case of facial expression analysis, the motion and deformation model can be taken from the shape and the motion characteristics of the head model description. In this context, a triangular B-spline model [6] is used to represent the face of a person. For rendering purposes, the continuous spline surface is discretized and approximated by a triangle mesh as shown in Fig. 2. The surface can be deformed by moving the splines' control points and thus affecting the shape of the underlying mesh. A set of MPEG-4 facial animation parameters characterizes the current facial expression and has to be estimated from the image sequence. By concatenating all transformations in the head model deformation and using knowledge from the perspective camera model, a relation between image displacements and FAPs can be analytically derived

$$\mathbf{d} = \mathbf{f}(\mathbf{FAP_0}, \mathbf{FAP_1}, \ldots, \mathbf{FAP_{N-1}}). \quad (2)$$

Combining this motion constraint with the optical flow constraint (1) leads to a linear systems of equations for the unknown FAPs. Solving this linear system in a least squares sense, results in a set of facial animation parameters that determines the current facial expression of the person in the image sequence.

B.2 Hierarchical Framework

Since the optical flow constraint (1) is derived assuming the image intensity to be linear, it is only valid for small motion displacements between two successive frames. To overcome this limitation, a hierarchical framework can be used [6]. First, a rough estimate of the facial motion and deformation parameters is determined from sub-sampled and low-pass filtered images, where the linear intensity assumption is valid over a wider range. The 3-D model is motion compensated and the remaining motion parameter errors are reduced on frames having higher resolutions.

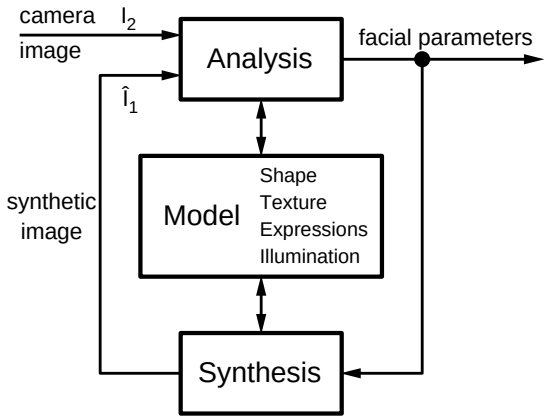

Fig. 4. Analysis-synthesis loop of the model-based estimator.

The hierarchical estimation can be embedded into an analysis-synthesis loop as shown in Fig. 4. In the analysis part, the algorithm estimates the parameter changes between the previous synthetic frame $\hat{I}$ and the current frame I' from the video sequence. The synthetic frame $\hat{I}$ is obtained by rendering the 3-D model (synthesis part) with the previously determined parameters. This approximate solution is used to compensate for the differences between the two frames by rendering the deformed 3-D model at the new position. The synthetic frame now approximates the camera frame much better. The remaining linearization errors are reduced by iterating through different levels of resolution. By estimating the parameter changes with a synthetic frame that corresponds to the 3-D model, an error accumulation over time is avoided.

C. Illumination Analysis

For natural video capture conditions, scene lighting often varies over time. This illumination variability not only has a considerable influence on the visual appearance of the objects in the scene, but also on the performance of computer vision algorithms or video coding methods. The efficiency and robustness of these algorithms can be significantly improved by removing the undesired effects of changing illumination. In our facial analysis and synthesis system, we model also lighting properties in the scene and estimate them. The explicit knowledge about light sources and reflection properties not only increases robustness of the facial expression analysis but also enables new possibilities for improvement or manipulation of lighting conditions. In the following, two ways of representing and estimating photometric properties in the scene are presented.

C.1 Explicit Light Source Models

The first approach [7] uses explicit models for light sources and surface reflection which are available for hardware rendering on common graphics cards. These models can also be described as nodes in the MPEG-4 BIFS scene and can easily be streamed to a client. In order to obtain a computationally efficient estimation of the parameters like color or direction of light, rather simple models are used that are sufficient to describe the dominant lighting effects in the scene. Lambertian reflection is assumed and the incoming light is modeled by a composition of a colored ambient component and a colored directional component with illuminant direction $\mathbf{l}$. The changes of the pixel color caused by illumination can be described by

$$\begin{aligned} I^R &= I^R_{tex}(c^R_{amb} + c^R_{dir} \cdot \max\{-\mathbf{n}\cdot\mathbf{l}, \mathbf{0}\}) \\ I^G &= I^G_{tex}(c^G_{amb} + c^G_{dir} \cdot \max\{-\mathbf{n}\cdot\mathbf{l}, \mathbf{0}\}) \\ I^B &= I^B_{tex}(c^B_{amb} + c^B_{dir} \cdot \max\{-\mathbf{n}\cdot\mathbf{l}, \mathbf{0}\}), \end{aligned} \tag{3}$$

where $[I^R_{tex} \;\; I^G_{tex} \;\; I^B_{tex}]^T$ denotes the pixel color originating from the texture map and $[I^R \;\; I^G \;\; I^B]^T$ their shaded correspondents. This equation has 8 unknowns (c^R_{amb}, c^G_{amb}, c^B_{amb}, c^R_{dir}, c^G_{dir}, c^B_{dir}, and two degrees of freedom for the normalized illuminant direction $\mathbf{l}$) that have to be estimated for each frame.

In contrast to the methods proposed in [1], [29], explicit surface normal information from the 3-D head model is available. This allows the usage of linear estimation techniques [7] which are computationally efficient. In the analysis-synthesis loop of Fig. 4 the lighting parameters are estimated in addition to motion and deformation in order to get a perfect match between synthesized and real camera frame. Similarly to the FAPs estimation, all pixels in the frame are considered to get robust estimates. For a detailed description of this method please refer to [7], [6].

C.2 Light Maps

An alternative approach to the explicit modeling of light sources and surface reflection is the superposition of several light maps which are attached to the object surface. Light maps are, similar to texture maps, two-dimensional images that are wrapped around the object containing shading instead of color information. During rendering, the unshaded texture map $I^C_{tex}(\mathbf{u})$ with $C \in \{R, G, B\}$ representing the three color components and the light map $L(\mathbf{u})$ are multiplied according to

$$I^C(\mathbf{u}) = \mathbf{I^C_{tex}}(\mathbf{u}) \cdot \mathbf{L}(\mathbf{u}) \tag{4}$$

in order to obtain a shaded texture map $I^C(\mathbf{u})$. The two-dimensional coordinate $\mathbf{u}$ specifies the position in both texture map and light map that are assumed to have the same mapping to the surface. For a static scene and viewpoint independent surface reflections, the light map can be computed off-line which allows the useage of more sophisticated shading methods as, e.g., radiosity algorithms [13] without slowing down the final rendering. This approach, however, can only be used if both object and light sources do not move. To overcome this limitation, we use a linear combination of scaled light maps instead of a single one

$$I^C(\mathbf{u}) = \mathbf{I^C_{tex}}(\mathbf{u}) \cdot \sum_{\mathbf{i=0}}^{\mathbf{N-1}} \alpha^{\mathbf{C}}_{\mathbf{i}} \mathbf{L_i}(\mathbf{u}). \tag{5}$$

By varying the scaling parameter α^C_i and thus blending between different light maps L_i, different lighting scenarios can be created. The N light maps $L_i(\mathbf{u})$ are again computed off-line with the same surface normal information $\mathbf{n}(\mathbf{u})$ but with different light source configurations. Fig. 5 shows an example of a 7 by 7 light map array.

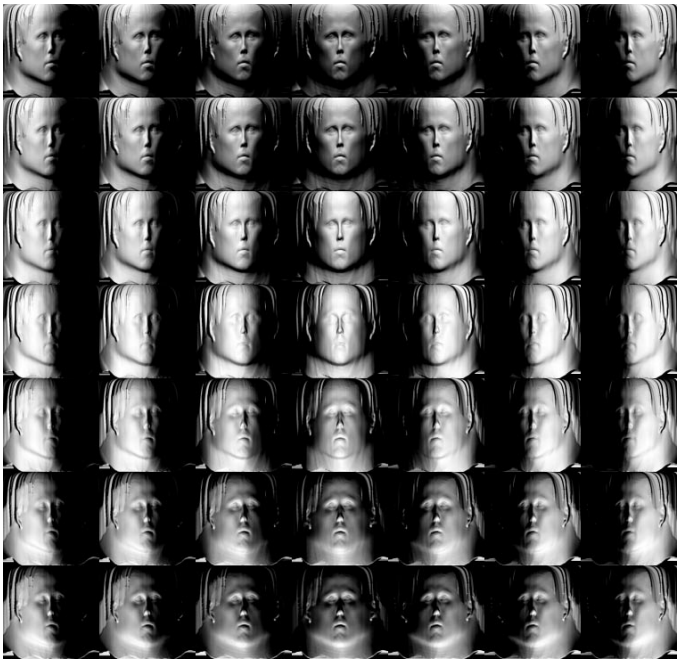

Fig. 5. Array of light maps for a configuration with 7 by 7 light sources.

In order to reduce the number of unknowns α^C_i that have to be estimated, a smaller orthogonal set of light maps is used rather than the original one. A Karhunen–Loève transformation (KLT) [30] is applied to the set of light maps L_i with $1 \le i \le N-1$ creating *eigen light maps* which concentrate most energy in the first representations. Hence, the number of degrees of freedom can be reduced without significantly increasing the mean squared error when reconstructing the original set. Fig. 6 shows the first three *eigen light maps* computed from the set of 49 different light maps.

For the lighting analysis of an image sequence, the parameters α^C_i have to be estimated for each frame. This is achieved in the same way as for the explicit light model analysis described in Section II-C.1. Only the shading model (3) is replaced by (5). A highly over-determined set of linear equations is set up and solved in a least squares sense for the unknown α^C_i's that represent all the photometric information. By varying α, lighting in the virtual scene can be adjusted or illumination variations in a real video sequence can be removed.

Fig. 6. First three *eigen light maps* representing the dominant shading effects.

D. Experimental Results

In this section, experimental results are presented for the proposed MPEG-4 facial analysis and synthesis system. Bit-rate measurements, image quality, and possible scene interactions and manipulations are shown for the applications of model-based coding, re-lighting, character animation, and view morphing.

D.1 Model-based Coding

Facial animation in video analysis and synthesis is especially suited for the encoding of head-and-shoulder image sequences at extremely low bit-rates. In model-based coding, 3-D analysis of facial expressions is used to animate a 3-D head model at the decoder as depicted in Fig. 2. The computer graphics scene description which is transmitted only once is identical at encoder and decoder. At the encoder side, the original camera frames are approximated by estimating motion and deformation parameters so that the virtual synthetic views match them optimally. After the initial scene description, only a few FAPs are streamed at extremely low bit-rates in order to update the dynamic scene changes.

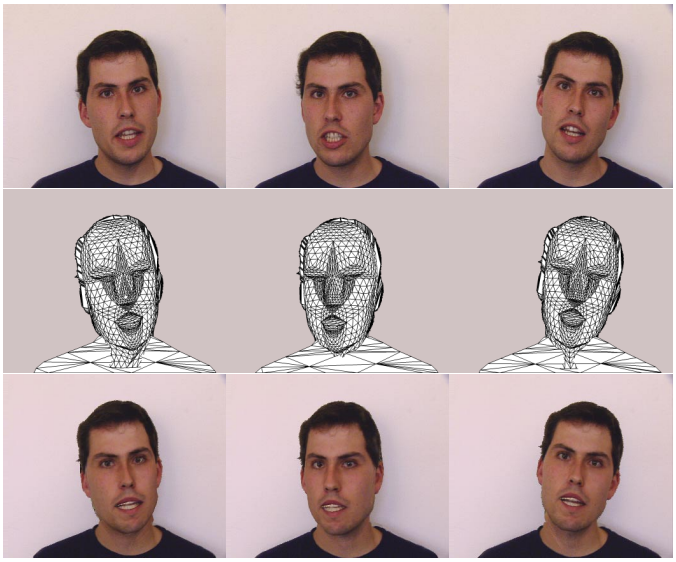

Fig. 7. Upper row: sequence *Peter*, middle row: animated wireframe model, lower row: synthesized model frames generated by rendering the 3-D head model.

In order to evaluate the performance of the model-based codec, a head-and-shoulder video sequence is recorded at CIF resolution (352 × 288) and 25 frames per second. In the top row of Fig. 7 three frames of this sequence are shown. The camera is calibrated using a model-based approach [5] and the generic head model is fitted to the first frame of the sequence by deforming the vertices. The frame is then projected onto the model, extrapolated, and used as texture map for the model. For all following frames, no model updates are performed except for deformations described by facial expression parameters. 22 of these parameters related to global pose, eye blinking, yaw, and lip movement are estimated at the encoder using the analysis-by-synthesis approach described in Section II-B. They are encoded and streamed to the decoder. Fig. 7 depicts three decoded frames (lower row) that are synthesized by rendering the 3-D head model which is illustrated in the middle row by means of a wireframe representation.

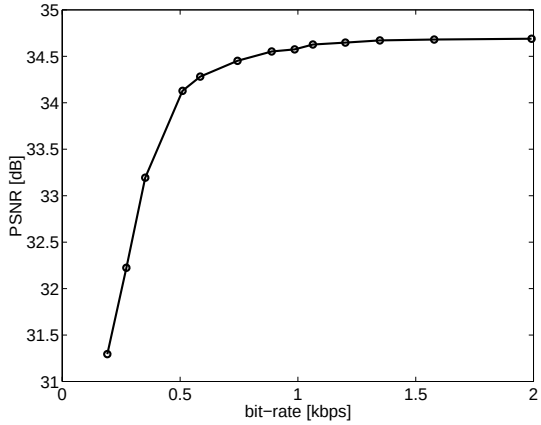

Fig. 8. PSNR versus bit-rate needed for FAP encoding.

Since only a few parameters are streamed over the network, extremely low bit-rates can be achieved. In this experiment, the 22 facial animation parameters are predicted from the previous frame and the prediction error is quantized and encoded with an arithmetic coder. By varying the quantizer, different bit-rates can be adjusted. Fig. 8 shows the peak-signal-to-noise-ratio (PSNR) measured only in the facial area (background is excluded) over the average bit-rate needed to encode the image sequence. The curve saturates at high bit-rates (at around 34.7 dB) since the 3-D head model provides only a limited accuracy. On the other hand, the reconstruction quality is degraded by less than 0.1 dB if the parameters are encoded at 1 kbit/s which corresponds to 40 bits per frame. When encoding only every third frame, the bit-rate is reduced to 0.5 kbit/s [4]. These low bit-rates enable the streaming of particular video sequences also for existing low data-rate communication channels as, e.g., wireless channels.

D.2 Modification of Scene Lighting

The explicit 3-D model description of all objects allows the application of different scene modifications and enhancements. One possibility is the change of illumination properties that effect the appearance of the image sequence. By adding explicit light sources or light maps to the virtual

scene as described in Section II-C the decoded sequences can be illuminated differently from the original. By tracking the head motion and retrieving surface normal information from the 3-D head model, also the original sequence can be manipulated by applying (3) or (5) to the camera images. This is illustrated in Fig. 9 where one frame of an image sequence is scaled with different parameters α_i leading to variable lighting.

Fig. 9. One frame of the sequence with different artificial lighting.

Vice versa, a video sequence recorded under varying lighting conditions can be recalculated to show constant illumination. This is illustrated in Fig. 10 where the upper row depicts some frames of the original sequence. During its capturing, several light sources are moved and switched on and off leading to severe intensity changes. Using the generic 3-D head model, the head motion and facial expressions are tracked over time. Since the texture of the model remains constant, the lighting variation in the original can be determined by estimating the α_i of the eigen light map approach described in Section II-C.2. The inverse of (5) is then applied to the original sequence to remove all lighting variations as shown in the lower row of Fig. 10. This could be used to enhance video sequences that are recorded under unfavorable lighting conditions or to remove these variations prior to their encoding with a hybrid video codec. Since hybrid video codecs rely on motion-compensated prediction, lighting changes cannot be handled efficiently. Video coding gains of about 2.9 dB in PSNR have been reported for an H.264 codec with this pre-processing step [8].

D.3 Expression Cloning

In the model-based coding system, the 3-D head models at encoder and decoder need not be the same. Simply by exchanging the model at the decoder, arbitrary people or artificial characters can be animated with the expressions of an actor at encoder side. Such an expression cloning [22] can be exploited in film productions for sensor-less motion capturing. An example is depicted in Fig. 11. The first row is the original that provides the facial motion and expression parameters. These are applied to 3-D models that are created from one single frame, respectively. The second and third row refer to animations rendered with these models with the same expression as in the top row. Besides film productions, this technique is also interesting for animating avatars or creating user friendly man machine interfaces.

Fig. 11. Expression cloning. Top row: original sequence that serves as input for the FAP estimator, middle and lower row: synthesized sequences with facial expressions extracted from the first row.

D.4 View Morphing

View morphing can be regarded as a further extension to expression cloning. Since the topology of the generic head model is the same for all individual 3-D representations, morphing between them can easily be accomplished by linearly displacing the vertices of the initial neutral expression model from the reference to the target description. Facial expression deformations can be applied all the time, not restricting the approach to static images. Moreover, the texture coordinates of all models are the same meaning that the texture maps are aligned pixel-wise. Therefore, the blending between the color can be realized by linearly blending between the texture map images. This view morphing is illustrated in Fig. 12. Both texture map and shape are blended linearly during the image sequence allowing the smooth switching between different models.

III. Model-Aided Coding

In model-based coding, the large class of head-and-shoulder image sequences can be represented and streamed very efficiently at bit-rates of a few kbit/s. This is possible, since a lot of a-priori knowledge about the objects in the scene is available that need not be encoded for each frame after an initial scene information exchange. However, only objects that can be represented by 3-D computer models

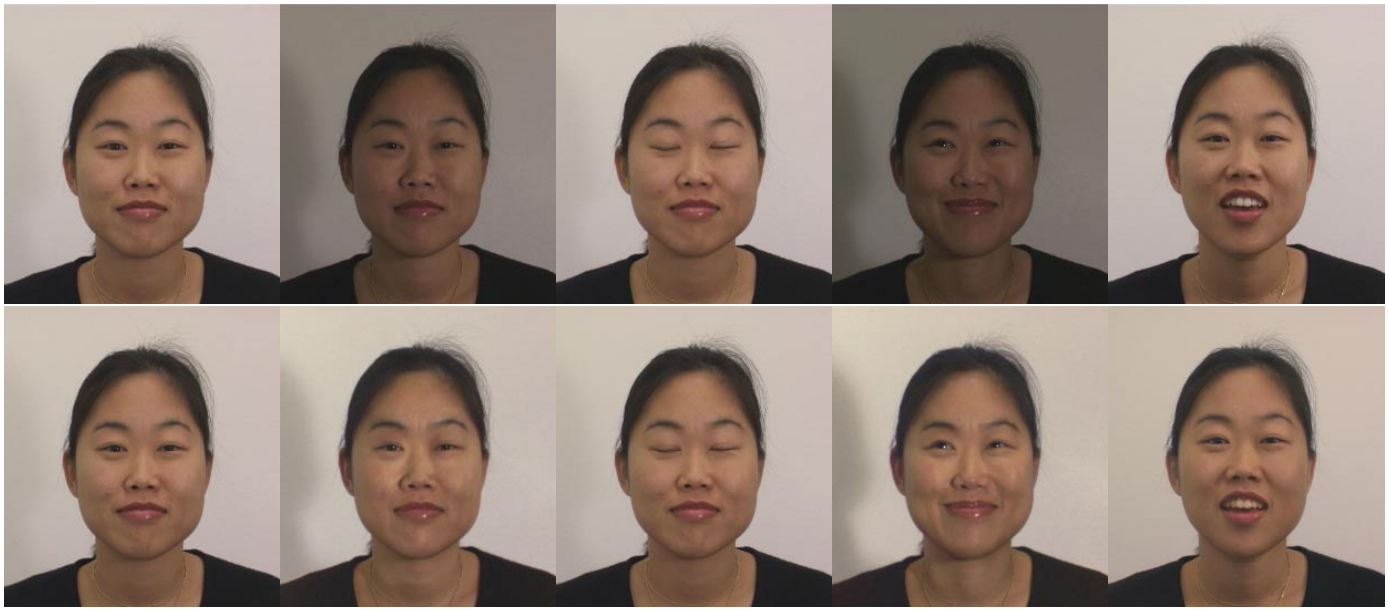

Fig. 10. Upper row: original sequence, lower row: corresponding illumination-compensated frames with constant lighting.

Fig. 12. View morphing between two different face models.

that are available at the decoder can be decoded. Unexpected objects like, e.g., a hand in front of the face simply do not show up unless they are additionally modeled. Researches have therefore worked on combining the efficiency of model-based codecs with the generality of hybrid video codecs like MPEG-4 [19] or H.264 [20]. Examples are the *switched model-based coder* [3] or the *layered coder* [21] which switch between the output of several different codecs frame-wise or object-wise. In [9], [10], we have proposed the *model-aided codec* (MAC) that uses the output of the model-based coder component only as a second reference image in multi-frame prediction. The mode-decision is performed for each block in the image and the additional residual coding and motion compensation of an already motion-compensated synthetic frame in combination with rate-distortion optimization lead to an efficient exploitation of the synthetic data. Moreover, the codec shows the same kind of artifacts for all image parts and avoid visually disturbing switching of different error patterns. If the 3-D models can describe the objects in the scene, high coding gains of several dB in PSNR can be achieved in comparison to the underlying hybrid codec. This model-aided coding concept has also shown its advantage in the efficient compression of 4-D light fields [18], [27] in image-based rendering.

A. Structure of the Model-Aided Codec

Fig. 13 shows the architecture of the proposed model-aided video codec. It depicts the well-known video coding loop that is extended by a model-based codec. The model-

based codec runs in parallel to the hybrid video codec, generating a synthetic model frame that is already motion-compensated. This model frame is employed as an additional reference frame for block-based motion-compensated prediction (MCP). For each block, the video coder decides which of the frames to use for MCP by minimizing a Lagrangian cost function $D + \lambda R$, where distortion D is minimized together with bit-rate R. The Lagrange parameter λ controls the balance between distortion and rate. The bit-rate reduction for the proposed scheme 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 residual transform coefficients is often dramatically reduced.

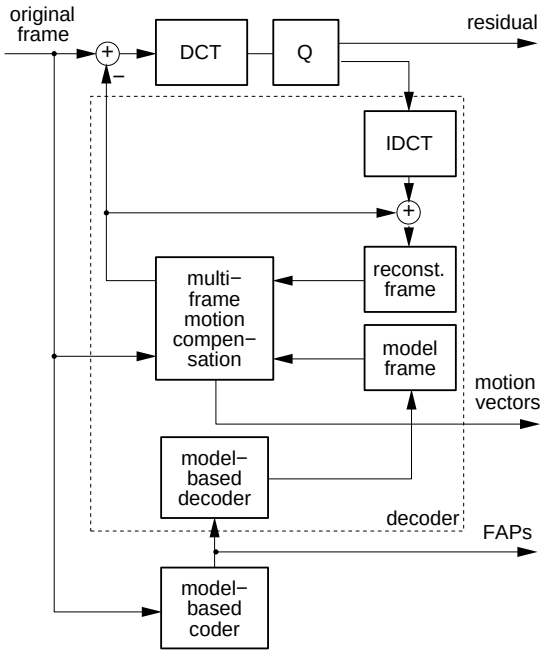

Fig. 13. Structure of the Model-Aided Codec.

In the experiments, an H.263 [15] and an H.264 [20] video codec are extended by the model-based component to build a model-aided codec. Multi-frame prediction must be enabled for both codecs and one of the reference frames is replaced by the synthesized model frame originating from the model-based coder.

B. Experimental Results

Experiments are conducted with two self-recorded natural CIF sequences *Clapper Board* and *Digital Camera*. The first sequence shows a typical head-and-shoulder scene of a speaking person with an additional clapper board object (see Fig. 14) that hides the face for the first 50 frames. Two scenarios are distinguished: the model-based codec is run with a second 3-D model of the clap for representing this object and without it relying on the residual coding of the underlying hybrid codec to reconstruct these image parts. The model-based system is embedded into an H.263+ codec (test model TMN-10) with annexes D, F, I, J, and T enabled. Motion compensation is performed on two frames, the previously decoded one and the synthesized model frame. Rate-distortion curves are measured by varying the DCT quantizer parameters over values 10, 15, 20, 25, and 31. Bit-streams are generated that are decodable producing the same PSNR values as at the encoder. The additional bit-rate for the facial animation parameters is considered while the data for the first INTRA frame and the 3-D model is excluded from the measurement simulating steady-state behavior. The results are compared to an H.263+ codec with the same options but with only one reference frame for prediction. The quality of the reconstructed frames for both cases is depicted in Fig. 14. Although the model-based codec has no 3-D model to synthesize the clap, it appears correctly at the MAC decoder which has a better visual quality than the corresponding frame from the H.263+ codec. Both frames originate from sequences encoded at the same average bit-rate of about 12 kbit/s and 8.33 Hz.

Fig. 14. Frame 54 of the sequence *Clapper Board*. Upper image: MAC: 36.1 dB, 2900 bits; lower image: TMN-10 with same bit-rate as MAC: 32.7 dB, 3200 bits.

The overall rate-distortion performance for the entire sequence is depicted in Fig. 15. At low bit-rates, bit-rate savings of about 33 % are achieved corresponding to coding gains of about 2.5 dB in PSNR at the same average bit-rate. If the clapper board is modeled by an additional 3-D object that is tracked with the same algorithm as described in Section II-B the efficiency of the MAC is fur-

ther improved, since more blocks can be predicted from the model frames. 45 % bit-rate reduction at the same quality or 3.5 dB increase in PSNR at the same average bit-rate are measured for the sequence *Clapper Board*.

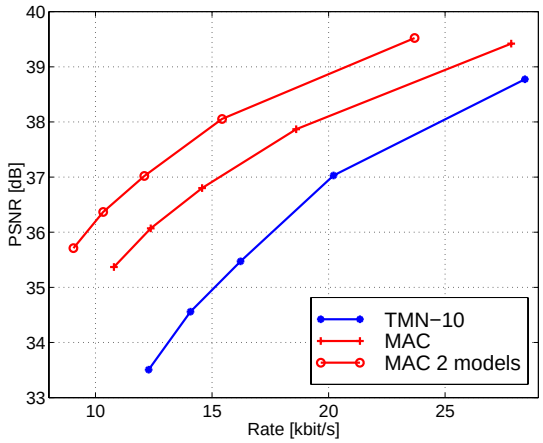

Fig. 15. PSNR versus bit-rate for sequence *Clapper Board*.

The model-aided codec is not restricted to head-and-shoulder scenes but any video sequence can be coded efficiently if 3-D models of at least some objects in the scene are available. This has been investigated in [2]. For this experiment, an H.264 [20] codec (test model TML-8) has been extended by a model-based codec. For the video sequence *Digital Camera* shown in Fig. 16, a simple approximate geometry model (right hand side of Fig. 16) is used. No texture is projected onto the 3-D description, but the shape is used to warp the previously decoded frame to the current position after having estimated the 3-D location of the object with the method described in Section II-B. Optionally, the background can additionally be warped using the 8-parameter motion model.

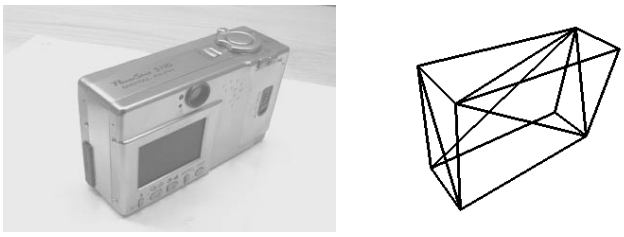

Fig. 16. Left: sequence *Digital Camera*, right: geometry model.

Fig. 17 shows the rate-distortion performance of the MAC in comparison to an H.264 codec running with one or two reference frames for prediction. 3 dB increase in PSNR at the same average bit-rate or , equivalently, 30 % bit-rate reduction at the same average quality is achieved. Further improvements can be obtained when modeling also the background with a second model.

The visual quality of the decoded frames is illustrated in Fig. 18. The frame on the top is decoded with an H.264 video codec while the lower one comes from the MAC, with both sequences encoded at the same average bit-rate. It can be observed that the use of 3-D models can support

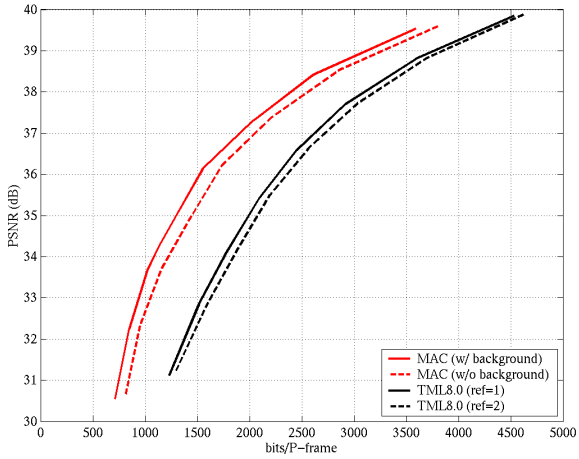

Fig. 17. PSNR versus bit-rate for sequence *Digital Camera*.

waveform coding and lead to improved visual quality even for imperfect modeling of the reality.

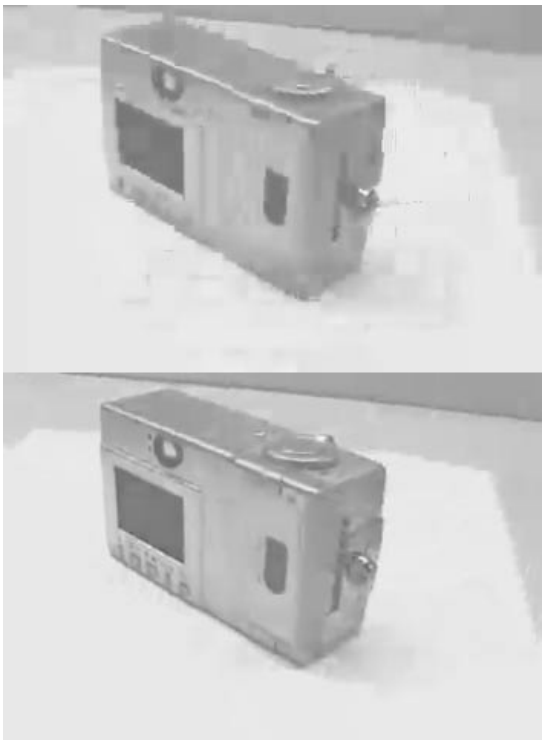

Fig. 18. Reconstructed frame 110 of sequence *Digital Camera*. Top: H.264: 31.1 dB, 1200 bits , bottom: MAC: 34.2 dB, 1200 bits.

IV. Conclusions

In this paper, an overview of our work on facial animation for video analysis and synthesis has been given. It has been shown that current techniques in model-based coding have reached the maturity for practical implementations in a wide variety of applications. Our system for facial expression analysis estimates MPEG-4 facial animation parameters using image information together with ex-

plicit knowledge from 3-D model descriptions. The linear algorithms that are embedded into a hierarchical analysis-synthesis loop together with the explicit consideration of illumination changes ensure robust estimates and fairly low computational complexity. Experiments have shown that head-and-shoulder image sequences can efficiently be encoded at bit-rates of about 1 kbit/s which makes such techniques useful for low bit-rate communication like wireless channels. Moreover, the semantic representation of the virtual scene allows the enhancement and modification of the image content enabling new applications like character animation, expression cloning, lighting enhancement, or view morphing. If pixel accurate representations are required, model-aided coding techniques can be used that combine the advantages of waveform coding and model-based coding. For video sequences with objects that can be modeled by 3-D computer models high bit-rate-reductions of 30-45 % at the same average quality can be achieved compared to state-of-the-art hybrid video codecs.

Acknowledgments

The author would like to thank Chuo-Ling Chang from Stanford University for providing Figures 16-18. Parts of this work have been accomplished by the author at the University of Erlangen, Germany, under the supervision of Prof. B. Girod, and at the Image, Video & Multimedia Systems Group, Stanford University. I would also like to thank Thomas Wiegand for contributing to the project on model-aided coding.

References

[1] 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 Transactions on Circuits and Systems for Video Technology*, 4(3):246–256, June 1994.

[2] C.-L. Chang, P. Eisert, and B. Girod. Using a 3D model for video coding. In *Proc. Vision, Modeling, and Visualization VMV'02*, pages 291–298, Erlangen, Germany, Nov. 2002.

[3] M. F. Chowdhury, A. F. Clark, A. C. Downton, E. Morimatsu, and D. E. Pearson. A switched model-based coder for video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):216–227, June 1994.

[4] P. Eisert. *Very Low Bit-Rate Video Coding Using 3-D Models*. PhD thesis, University of Erlangen, Shaker Verlag, Aachen, Germany, 2000.

[5] P. Eisert. Model-based camera calibration using analysis by synthesis techniques. In *Proc. Vision, Modeling, and Visualization VMV'02*, pages 307–314, Erlangen, Germany, Nov. 2002.

[6] P. Eisert and B. Girod. Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, 18(5):70–78, Sep. 1998.

[7] 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*, pages 119–122, Alpbach, Austria, Jul. 1998.

[8] P. Eisert and B. Girod. Model-based enhancement of lighting conditions in image sequences. In *Proc. SPIE Visual Communications and Image Processing, VCIP-02*, San Jose, USA, Jan. 2002.

[9] P. Eisert, T. Wiegand, and B. Girod. Rate-distortion-efficient video compression using a 3-D head model. In *Proc. International Conference on Image Processing (ICIP)*, volume 4, pages 217–221, Kobe, Japan, Oct. 1999.

[10] P. Eisert, T. Wiegand, and B. Girod. Model-aided coding: A new approach to incorporate facial animation into motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(3):344–358, Apr. 2000.

[11] R. Forchheimer, O. Fahlander, and T. Kronander. Low bit-rate coding through animation. In *Proc. Picture Coding Symposium (PCS)*, pages 113–114, Davis, California, Mar. 1983.

[12] M. Frey, P. Giovanoli, H. Gerber, M. Slameczka, and E. Stüssi. Three-dimensional video analysis of facial movements: A new method to assess the quantity and quality of the smile. *Plastic and Reconstructive Surgery*, 104(7):2032–2039, Dec. 1999.

[13] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. In *Proc. Computer Graphics (SIGGRAPH)*, volume 18, pages 213–222, Jul. 1984.

[14] P. Hammond, T. J. Hutton, M. A. Patton, and J. E. Allanson. Delineation and visualisation of congenital abnormality using 3D facial images. In *Intelligent Data Analysis in Medicine and Pharmacology*, London, UK, Sep. 2001.

[15] ITU-T Recommendation H.263 Version 2 (H.263+). Video Coding for Low Bitrate Communication. Jan. 1998.

[16] G. A. Kalberer and L. Van Gool. Lip animation based on observed 3D speech dynamics. In *Proc. Visual Computation and Image Processing (VCIP)*, pages 16–25, Jan. 2001.

[17] P. Kauff and O. Schreer. Virtual team user environments - a step from tele-cubicles towards distributed tele-collaboration in mediated workspaces. In *Proc. International Conference on Multimedia and Expo (ICME)*, Lausanne, Switzerland, Aug. 2002.

[18] M. Magnor, P. Eisert, and B. Girod. Multi-view image coding with depth maps and 3-D geometry for prediction. In *Proc. Visual Computation and Image Processing (VCIP)*, San Jose, USA, Jan. 2001.

[19] *ISO/IEC FDIS 14496-2, Generic Coding of audio-visual objects: (MPEG-4 video), Final Draft International Standard, Document N2502*, 1999.

[20] *Joint Video Specification (ITU-T Rec. H.264 — ISO(IEC 14496-10 AVC), Draft, Document JVT-E022d7*, Sep. 2002.

[21] H. G. Musmann. A layered coding system for very low bit rate video coding. *Signal Processing: Image Communication*, 7(4-6):267–278, Nov. 1995.

[22] J.-Y. Noh and U. Neumann. Expression cloning. In *Proc. Computer Graphics (SIGGRAPH)*, Los Angeles, USA, Aug. 2001.

[23] F. I. Parke and K. Waters. *Computer Facial Animation*. A K Peters, Massachusetts, 1996.

[24] D. E. Pearson. Developments in model-based video coding. *Proceedings of the IEEE*, 83(6):892–906, June 1995.

[25] R. W. Picard. *Affective Computing*. MIT Press, Cambridge, USA, 1997.

[26] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proc. Computer Graphics (SIGGRAPH)*, pages 75–84, Orlando, Florida, Jul. 1998.

[27] P. Pramanath, E. Steinbach, P. Eisert, and B. Girod. Geometry refinement for light field compression. In *Proc. International Conference on Image Processing (ICIP)*, Rochester, USA, Sep. 2002.

[28] M. Rydfalk. *CANDIDE: A Parameterized Face*. PhD thesis, Linköping University, 1978. LiTH-ISY-I-0866.

[29] J. Stauder. Estimation of point light source parameters for object-based coding. *Signal Processing: Image Communication*, 7(4-6):355–379, Nov. 1995.

[30] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal for Cognitive Neuroscience*, 3(1):71–86, 1991.

[31] T. Vetter and V. Blanz. Estimating coloured 3D face models from single images: An example based approach. In *Proc. European Conference on Computer Vision (ECCV)*, volume 2, pages 499–513, Freiburg, Germany, June 1998.

[32] W. J. Welsh, S. Searsby, and J. B. Waite. Model-based image coding. *British Telecom Technology Journal*, 8(3):94–106, Jul. 1990.