EURASIP Journal on Applied Signal Processing 2002:6, 566–571
© 2002 Hindawi Publishing Corporation

An Active Model for Facial Feature Tracking

Jörgen Ahlberg
Image Coding Group, Department of Electrical Engineering, Linköping University, SE-581 31 Linköping, Sweden
Email: ahlberg@isy.liu.se

Received 7 July 2001 and in revised form 5 March 2002

We present a system for finding and tracking a face and extract global and local animation parameters from a video sequence. The system uses an initial colour processing step for finding a rough estimate of the position, size, and inplane rotation of the face, followed by a refinement step drived by an active model. The latter step refines the previous estimate, and also extracts local animation parameters. The system is able to track the face and some facial features in near real-time, and can compress the result to a bitstream compliant to MPEG-4 face and body animation.

Keywords and phrases: active appearance models, free tracking, facial feature tracking, model-based coding, MPEG-4 face and body animation.

1. INTRODUCTION

The goal of this work is to extract parameters describing the adaptation of a face model to the frames of a video sequence. Since the target applications are video-phones and man-machine interaction, we assume a "video phone-like" input, that is, we assume that there is a face in the image, looking approximately into the camera.

We use the face model Candide-3 [1], and the adaptation parameters as global (rotation, translation, scale) as well as local (action units [1, 2] controlling the mouth and the eyebrows).

Initially, a colour-based algorithm, assuming that skin colour is recognizable, is used to give a rough estimate of the size and position of the face. Those parameters are then refined by an active model. The two algorithms are presented in the subsequent sections, followed by the results, and a description of our ongoing work.

2. BACKGROUND: ACTIVE APPEARANCE MODELS

Deformable models describing the shape of a class of objects (e.g., faces) have been in use for some time, evolving from the *snakes* or *active contour models* introduced in 1988 [3] to statistical models like *point distribution models* (PDMs) and *active shape models* (ASMs) [4, 5]. The shape of a statistical shape model is typically controlled by adding a linear combination of shape/deformation modes to an average shape $\bar{\mathbf{s}}$ according to

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{S}\sigma, \tag{1}$$

where σ is a vector of shape/deformation parameters, and the columns of $\mathbf{S}$ contains the deformation modes. The resulting vector $\mathbf{s}$ contains the 2D or 3D coordinates of the model's control points, edge points, or vertices.

In parallel, statistical models describing the texture of objects have been developed, starting with Turk and Pentland introducing *eigenfaces* for face recognition in 1991 [6]. The main problem with the eigenfaces soon turned out to be that the face distribution in the image space is highly nonconvex and complex, and thus it is not well modelled by a linear basis. The two popular ways of handling this problem are to create either a nonlinear model, for example, by ISOMAP [7] or LLE [8], or *geometrically normalized eigenfaces*, introduced by Ström et al. who used it for image compression [9]. Geometrically normalized eigenfaces are also called *shape free* eigenfaces, *eigentextures*, or *texture modes*. The parameterized (and geometrically normalized) texture of a model is described by

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{X}\xi, \tag{2}$$

where the texture parameters ξ put weights on the eigentextures (columns of $\mathbf{X}$) and $\bar{\mathbf{x}}$ is the average texture.

Appearance models [10] combine shape and texture into one statistical model, where a number of *appearance modes* are controlled by the model parameters. Such a model would consequently be expressed as

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{A}_s\mathbf{p}, \qquad \mathbf{x} = \bar{\mathbf{x}} + \mathbf{A}_x\mathbf{p}, \tag{3}$$

that is, the same parameters control shape as well as texture.

In 1998, the *active appearance models* (AAMs) were introduced by Edwards et al. [10], combining an appearance model with a fast search algorithm for matching the model to an image. The AAMs have been the target for quite some research since, and extensions include view-based face spaces

[11], modelling the ageing of faces [12], face detection [13], and constrained AAM search [14], to pick a few. A comparison between ASMs and AAMs can be found in [15], and a combination in [16]. For an introduction to AAMs, see [17]. A more in-depth treatment can be found in [18]. The AAM search algorithm will be briefly described below.

2.1. The AAM search algorithm

Given an input image $\mathbf{i}$ and a model parameter vector $\mathbf{p}$, we can map the image onto the model, reshape the model to the standard shape, and thus create a normalized input image

$$\mathbf{j} = \mathbf{j}(\mathbf{i}, \mathbf{s}(\mathbf{p})). \tag{4}$$

From now on we fix the input image and regard $\mathbf{j}$ as a function of $\mathbf{p}$ only. We compute the residual image as

$$\mathbf{r}(\mathbf{p}) = \mathbf{j}(\mathbf{p}) - \mathbf{x}(\mathbf{p}), \tag{5}$$

and our goal is to find the optimal adaptation of the model to the input image, that is, to find the $\mathbf{p}$ that minimizes the error measure

$$e(\mathbf{p}) = \|\mathbf{r}(\mathbf{p})\|^2. \tag{6}$$

Assuming that we have a rough estimate of the model parameters, we would like to find the update vector $\Delta\mathbf{p}$ that minimizes $e(\mathbf{p} + \Delta\mathbf{p})$. Following [17], we Taylor-expand $\mathbf{r}$ around $\mathbf{p} + \Delta\mathbf{p}$ getting

$$\mathbf{r}(\mathbf{p} + \Delta\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \mathbf{G}\Delta\mathbf{p} + O(\Delta\mathbf{p}), \tag{7}$$

where

$$\mathbf{G} = \frac{\partial}{\partial \mathbf{p}}\mathbf{r}(\mathbf{p}). \tag{8}$$

We approximate $\mathbf{r}(\mathbf{p})$ with the first terms and regard it as a linear function. Thus, we want to minimize

$$e(\mathbf{p} + \Delta\mathbf{p}) = \|\mathbf{r}(\mathbf{p}) + \mathbf{G}\Delta\mathbf{p}\|^2 \tag{9}$$

of which the least square solution is

$$\Delta\mathbf{p} = \mathbf{U}\mathbf{r}(\mathbf{p}) = -(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{r}(\mathbf{p}). \tag{10}$$

The vector $\Delta\mathbf{p}$ gives us a probable update in the search space, and we compute a new parameter vector and a new error measure

$$\mathbf{p}' = \mathbf{p} + \Delta\mathbf{p}, \qquad e' = e(\mathbf{p}'). \tag{11}$$

If $e' < e$, we update $\mathbf{p}$ accordingly ($\mathbf{p}' \rightarrow \mathbf{p}$) and iterate until convergence. If $e' > e$, we try smaller update steps (0.5 and 0.25). If neither of these improves the error measure, we declare convergence.

2.2. Training the AAM

From a set of training data (models adapted to images), we can estimate the gradient matrix $\mathbf{G}$, and from the estimated $\mathbf{G}$ we compute the update matrix $\mathbf{U}$ as the negative pseudoinverse of $\mathbf{G}$:

$$\mathbf{U} = -\mathbf{G}^{\dagger} = -(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T. \tag{12}$$

To be able to use the AAM search, we should consequently estimate the gradient matrix $\mathbf{G}$. We do that by perturbating $\mathbf{p}$ from the set of (manually) adapted models, parameter by parameter, step by step. The jth row in $\mathbf{G}$ can thus be estimated as

$$\mathbf{G}_j = \sum_k \left(\mathbf{r}(\mathbf{p} + \Delta\mathbf{p}_{jk}) - \mathbf{r}(\mathbf{p})\right), \tag{13}$$

where $\Delta\mathbf{p}_{jk}$ is a vector that perturbs $\mathbf{p}$ in the jth component to the amount of $k \cdot c$ for some suitable constant c.

3. OUR ACTIVE MODEL AND ITS PARAMETERIZATION

Our active model is a simplification of the AAM, and we describe here the model and how it is parameterized. As a starting point, we use the wireframe face model Candide, which has been popular in video coding research for many years. The third variant of this model, Candide-3, is also compliant to MPEG-4 Face Animation.

The Candide model has manually been adapted to a set of images by varying a set of 12 parameters:

- the first six parameters are the global motion (or pose) parameters; 3D-rotation, 2D-translation, and scale. We denote the pose parameter vector $\pi = [r_x, r_y, r_z, t_x, t_y, z]^T$;
- the remaining parameters are activation levels for six action units (from FACS, the facial action coding system [2]) controlling the lips and eyebrows. They are contained in the vector σ, and control the shape of the head according to (1).

We collect those parameters in a 12D vector $\mathbf{p}^T = [\pi^T, \sigma^T]$, which parameterizes the geometry of the model. Thus, the geometry is described by

$$\mathbf{g}(\mathbf{p}) = \mathbf{R}(z+1)(\bar{\mathbf{s}} + \mathbf{S}\sigma) + \tau, \tag{14}$$

where the columns of $\mathbf{S}$ are action unit definitions from Candide-3, σ is the vector of action unit activation levels, $\bar{\mathbf{s}}$ is the standard shape of the Candide model, $\tau = \tau(t_x, t_y)$ is a function of the x- and y-translations, $\mathbf{R} = \mathbf{R}(r_x, r_y, r_z)$ is a rotation matrix created from the three rotation parameters, and z is a scaling factor.

The image under the model has, for each image in the training set, been mapped onto the model, creating a texture mapped wireframe model. The model has then been normalized to a standard shape, size, and position (i.e., $\mathbf{p} = 0 \Rightarrow \mathbf{g} = \bar{\mathbf{g}}$), in order to collect a geometrically normalized set of textures. On this set, a PCA has been performed and the eigentextures (shape free eigenfaces, geometrically normalized eigenfaces) have been computed.

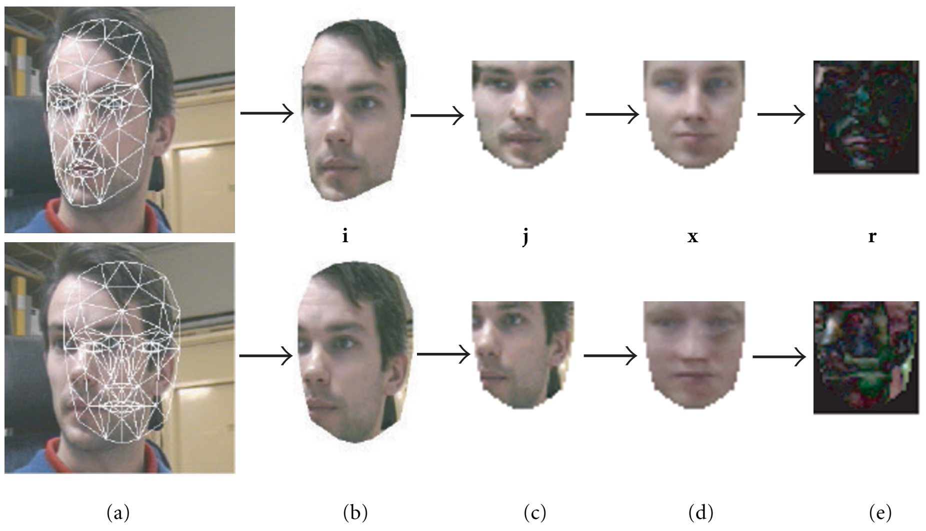

FIGURE 1: The model matching and texture approximation process. (a) A good and a bad (top and bottom row, respectively) model adaptation is shown; (b) the image mapped onto the model; (c) the model is reshaped to the standard shape, producing the image **j**; (d) the normalized texture is approximated by the eigentextures, producing the image **x**; (e) the residual image **r** is computed. The better the model adaptation is the more similar the images **j** and **x** are. Analysis of the image **r** tells us how to improve the model adaptation, that is, how to minimize the difference between **j** and **x**.

We can now describe the complete appearance of the model by the geometry parameters **p** and an N-dimensional texture parameter vector ξ, where N is the number of eigentextures we want to use for synthesizing the model texture.

3.1. Matching the model and the image

As in (4) we can create the normalized input image $\mathbf{j}(\mathbf{p})$. The texture parameters minimizing the model error $e(\mathbf{p})$ are then given by projecting the normalized input image $\mathbf{j}(\mathbf{p})$ on the eigentextures, that is,

$$\xi = \mathbf{X}^T(\mathbf{j}(\mathbf{p}) - \bar{\mathbf{x}}). \quad (15)$$

Thus, **p** is the only necessary parameter in our case:

$$\mathbf{g} = \mathbf{g}(\mathbf{p}), \qquad \mathbf{x} = \mathbf{x}(\mathbf{p}), \quad (16)$$

where

$$\mathbf{x}(\mathbf{p}) = \bar{\mathbf{x}} + \mathbf{X}\mathbf{X}^T(\mathbf{j}(\mathbf{p}) - \bar{\mathbf{x}}). \quad (17)$$

The entire process from input image to normalized is illustrated in Figure 1.

Note that our model is not a complete appearance model, since we parameterize the geometry only and let the texture depend on the input image as well. For an appearance models, a PCA is performed to find the suitable subspace of appearance modes combining deformation modes and texture modes (eigentextures). In our application, we only parameterize the model in terms of deformation (including global motion) since we know in advance what kind of parameters we are interested in extracting. If we want to extract action units (the parameters typically used for CANDIDE), we simply parameterize and train our model on those parameters (or deformations spanning the same subspace). We can still use the AAM search exactly as described above.

3.2. Training the model

To try out this scheme, the model has been adapted (manually in the beginning, then semiautomatically) to 330 images of six different persons from different angles and with different facial expressions. The following action units from FACS have been chosen as deformation parameters:

(1) jaw drop;
(2) lip stretcher;
(3) lip corner depressor;
(4) upper lip raiser;
(5) eyebrow lowerer;
(6) outer eyebrow raiser.

The adapted model, for each image, has been normalized to a standard shape with the size 40 × 42 pixels, as in Figure 1c, top row, and a PCA has been performed on the resulting training textures to compute the eigentextures. The mean texture $\bar{\mathbf{x}}$ and the DC-level have been subtracted from the training textures prior to the PCA.

With the eigentextures available, all the parameters have been perturbed, one by one and for each image, in steps of 0.01 in the range $[-0.1, 0.1]$, and the matrix **G** estimated. From **G**, the update matrix **U** has been computed.

3.3. Using the active model for tracking

The AAM search efficiently adapts the model to the image provided that the initial estimate of the model parameters are good enough. The better the initial estimate is, the smaller is the risk of the AAM search getting stuck in a local minimum

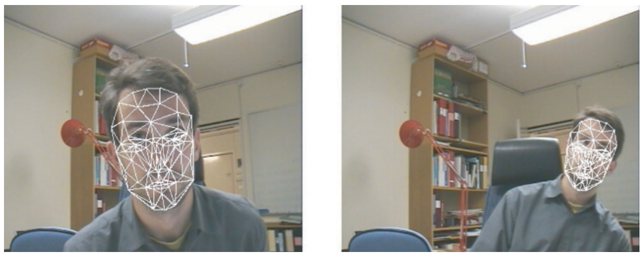

Figure 2: Examples of the colour-based algorithm giving a rough initial estimate of the location, size, and inplane rotation of the face.

not corresponding to the correct model adaptation. In a video sequence, the changes between each frame are quite small (provided that the frame rate is high enough) so the adaptation from the previous frame can be used as the initial estimate. However, for the first frame, some other technique has to been used, as discussed in Section 4. Also note that since we do not change the static shape, the model should be adjusted to the subject feature's in advance, using the shape units.

4. THE INITIAL STEP: THE COLOUR-BASED ALGORITHM

To quickly find the approximate location of the face in the input image, the pixels are traversed and each is given a likelihood value of being a skin coloured pixel. This likelihood value is based on a priori collected statistics, to which a mixture of Gaussian distributions has been adapted using the expectation-maximization (EM) algorithm. The resulting image of likelihood values is blurred and then thresholded, and of the remaining objects in the image, the largest one is selected as the most probable face candidate. The position, size, and orientation of this "blob" is used as the initial estimate handed over to the refinement step. Examples of resulting estimates are shown in Figure 2.

This kind of algorithm has been chosen because it is fast and simple. The obvious drawback is that it needs recalibration for each camera and for differing lighting conditions.

5. IMPLEMENTATION

We have implemented a C++ library with routines for handling face models and training them as to be active models. The shape of the (normalized) eigentextures is determined by the standard shape of the Candide model (with the upper part of the head removed) scaled so that the size is 40×42 pixels (see Figures 1c and 1d). The eigentextures have been computed in RGB as well as in grayscale for comparison, the eigentextures and the update matrix $\mathbf{U}$ have been computed in RGB as well as in grayscale.

The implementation uses OpenGL for the texture mapping in the AAM search, utilizing the fact that modern graphics cards have specialized hardware for such tasks. The geometrical normalization of the input image (see Figure 1c) is thus performed in a very short time (less than 2 milliseconds), and the speed of the algorithm is dependent more on the graphics card than on the CPU.

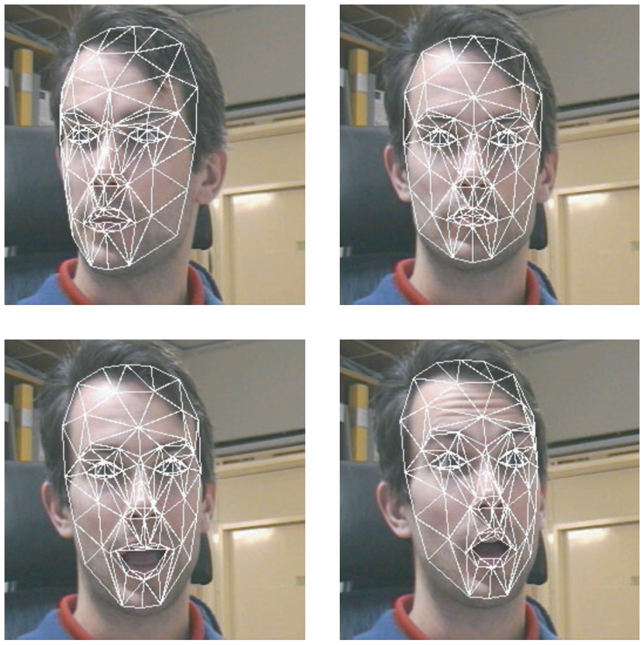

Figure 3: The Candide-3 model adapted to four frames of a video sequence.

5.1. MPEG-4 encoding

The animation of the face model can be encoded using the MPEG-4 standard for face animation. Since the vertices of the Candide-3 model correspond well to the facial feature points defined in MPEG-4, the coding is easily done. Details on how to compute the Facial Animation Parameters (FAPs) from Candide-3 can be found in [1].

The FAPs used to represent movements of the facial feature points in MPEG-4 are measured in face dependent scales, using different FAP Units (FAPUs). The FAPs are also measured relative to the neutral face, and thus a neutral face model is kept in memory. Using this neutral face model, the FAPUs and the FAPs are computed, and then compressed using the MPEG-4 reference software. The entire process takes only about five milliseconds per frame and does not influence the real-time performance. The output is an MPEG-4 Face and Body Animation (FBA) compliant bitstream, that can be played in an FBA player, for example, the Facial Animation Engine (FAE) [19] or MpegWeb [20]. The bitstream can be stored on a file or streamed over the network.

6. RESULTS

The experiments presented here are performed on a PC with a 500 MHz Intel Pentium III processor and an ASUS V3800 graphics card with video input. The colour-based algorithm runs on approximately 0.1 seconds, and the AAM search needs about 15 ms/iteration. Typically, less than 10 iterations are needed each frame, and fewer iterations are needed the

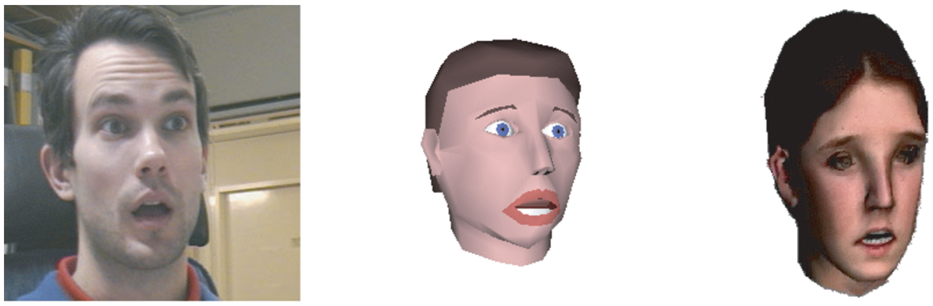

Figure 4: The original frame (left), the frame synthesized with FAE from DIST, University of Genova (middle), and the facial animation system by Miralab, University of Geneva (right). The corresponding Candide model adaptation is shown in Figure 3 (right).

Table 1: Timing results (average over 341 frames).

Measurement	RGB	Grayscale
Iterations per frame	6.9	6.8
Total time per frame (ms)	94.1	69.1
Time per iteration (ms)	13.6	10.2
Time for computing $\Delta\mathbf{p}$ (ms)	7.2	5.05

closer the initial estimate is to the optimum. Thus, if a video sequence is recorded at a high frame rate (with small motion between each frame), the tracking will also run on a higher speed. Visual results are shown in Figure 3.

Using grayscale eigentextures and update data, it turned out that the computation in the graphics card (which internally uses RGB) became almost 20% slower. However, the computations performed in the CPU became (as expected) about 3 times faster, and then only 20% of the total computing time is due to the CPU (the rest being computations in the graphics card).

Testing on a video sequence of a few hundred frames gave results according to Table 1. It is clear that the grayscale computations are preferable, since the visual results are equivalent.

Snapshots from the animation of the resulting MPEG-4 Face Animation bitstream using two different facial animation systems are shown in Figure 4.

7. CURRENT WORK AND FUTURE IMPROVEMENTS

There are several ways this system can be improved, and they are currently under investigation. Four things to be considered are mentioned here.

First, the colour-based algorithm is not robust enough, and it should be complemented with some more simple and fast technique. For example, we could require that an area could be a face candidate only if there is some difference (due to motion) between the first and second frame.

Second, $\mathbf{U}$ is a somewhat sparse matrix. By utilizing this fact, the computation time could be improved.

Third, as all tracking systems, this system can lose track, and therefore, some kind of re-initialization scheme is needed. One possible procedure is that when the active model does not converge to a small error measure, the colour-based algorithm is invoked, handing a new initial estimate to the active model. Re-initialization issues are discussed in [21].

Fourth, our currently used set of action units is not complete. For example, we do not analyse the motion of the eyelids at all, and the shape of the head is assumed to be known a priori.

8. CONCLUSION

We have presented a system that tracks a face and facial features in a video sequence. The resulting animation data is encoded using MPEG-4 Face Animation. The system works in near real-time, and the experimental results are promising. With some further development and optimization, a real-time 3D face and facial feature tracker should be possible to implement on consumer hardware.

ACKNOWLEDGMENTS

This work is financed by the European IST project InterFace and the national Swedish SSF project VISIT. The author is also thankful to Dr. Tim Cootes for advice on the AAM training.

REFERENCES

[1] J. Ahlberg, "CANDIDE-3—an updated parameterized face," Report No. LiTH-ISY-R-2326, Department of Electrical Engineering, Linköping University, Sweden, 2001.

[2] P. Ekman and W. V. Friesen, *Facial Action Coding System*, Consulting Psychologist Press, Palo Alto, Calif, USA, 1977.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[4] T. F. Cootes and C. J. Taylor, "Active shape models—'smart snakes'," in *Proc. British Machine Vision Conference*, pp. 266–275, Leeds, UK, September 1992.

[5] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, PWS Publishing, Pacific Grove, Calif, USA, 2nd edition, 1999.

[6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global

geometric framework, for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[8] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[9] J. Ström, F. Davoine, J. Ahlberg, H. Li, and R. Forchheimer, "Very low bit rate facial texture coding," in *Proc. Int. Workshop on Synthetic/Natural Hybrid Coding and 3D Imaging*, pp. 237–240, Rhodes, Greece, September 1997.

[10] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Interpreting face images using active appearance models," in *Proc. 3rd Int. Conf. on Automatic Face and Gesture Recognition*, pp. 300–305, Nara, Japan, 1998.

[11] T. F. Cootes, K. N. Walker, and C. J. Taylor, "View-based active appearance models," in *Proc. Int. Conf. on Face and Gesture Recognition*, pp. 227–232, Grenoble, France, 2000.

[12] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Modeling the process of ageing in face images," in *Proc. Int. Conf. on Computer Vision*, vol. 1, pp. 131–136, Kerkyra, Greece, 1999.

[13] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Advances in active appearance models," in *Proc. Int. Conf. on Computer Vision*, vol. 1, pp. 137–142, Kerkyra, Greece, 1999.

[14] T. F. Cootes and C. J. Taylor, "Constrained active appearance models," in *Proc. Int. Conf. Computer Vision*, vol. 1, pp. 748–754, Vancouver, Canada, 2001.

[15] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Comparing active shape models with active appearance models," in *Proc. British Machine Vision Conference*, vol. 1, pp. 173–182, Nottingham, UK, 1999.

[16] S. Mitchell, B. Lelieveldt, R. van der Geest, J. Shaap, J. Reiber, and M. Sonka, "Segmentation of cardiac MR images: an active appearance model approach," in *Proc. SPIE Medical Imaging*, San Diego, Calif, USA, February 2000.

[17] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[18] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision," Draft report, Wolfson Image Analysis Unit, University of Manchester, February 2001.

[19] F. Lavagetto and R. Pockaj, "The facial animation engine: towards a high-level interface for the design of MPEG-4 compliant animated faces," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 2, pp. 277–289, 1999.

[20] I. S. Pandzic, "MpegWeb—an MPEG-4 facial animation applet," web-based software at http://www.icg.isy.liu.se/~igor/MpegWeb.

[21] J. Ström, "Reinitialization of a model-based face tracker," in *Proc. Int. Conf. on Augmented, Virtual Environments and 3-D Imaging*, pp. 128–131, Mykonos, Greece, 2001.

Jörgen Ahlberg was born in 1971 in Karlstad, Sweden. He received his M.S. in computer science and engineering at Linköping University, Sweden, in 1996, and joined the Image Coding Group as a Ph.D. student in the same year. Since then, he has been undertaking research in compression of face model parameters for model-based coding, facial feature extraction, evaluation of face models, and face and facial feature tracking. He was also an active participant in the development of MPEG-4 Face Animation. In 1999, he was a visiting researcher at Université de Technologie de Compiègne, France. Currently he is finishing his Ph.D. studies at the Image Coding Group.