LAFTER: Lips and Face Real Time Tracker

Nuria Oliver, Alex P. Pentland
Vision And Modeling Group
MIT Media Laboratory
Cambridge, MA 02139, USA
{nuria,sandy}@media.mit.edu

François Bérard
CLIPS-IMAG, BP 53
38041 Grenoble cedex 9 France
francois.berard@imag.fr

Abstract

This paper describes an active-camera real-time system for tracking, shape description, and classification of the human face and mouth using only an SGI Indy computer. The system is based on use of 2-D blob features*, which are spatially-compact clusters of pixels that are similar in terms of low-level image properties. Patterns of behavior (e.g., facial expressions and head movements) can be classified in real-time using Hidden Markov Model (HMM) methods. The system has been tested on hundreds of users and has demonstrated extremely reliable and accurate performance. Typical classification accuracies are near 100%.*

1. Introduction

This paper describes a real-time system for accurate tracking and shape description, and classification of the human face and mouth using 2-D *blob features* and Hidden Markov Models (HMMs). All of the experimental apparatus described here is real-time, at 20 to 30 frames per second, and runs on SGI Indy workstations without any special-purpose hardware.

The notion of "blobs" as a representation for image features has a long history in computer vision [25, 19, 5, 34], and has had many different mathematical definitions. In our usage it is a compact set of pixels that share a visual property that is not not shared by the surrounding pixels. This property could be color, texture, brightness, motion, shading, a combination of these, or any other salient spatio-temporal property derived from the signal (the image sequence). In our usage blobs are, therefore, a coarse, locally-adaptive encoding of the images' spatial and color/texture/motion/etc. properties. A prime motivation for our interest in blob representations is our discovery that they can be reliably detected and tracked even in complex, dynamic scenes, and that they can be extracted in real-time without the need for special purpose hardware. These properties are particularly important in applications that require tracking people, and recently we have used 2-D blob tracking for real-time whole-body human interfaces [34] and real-time recognition of American Sign Language hand gestures [32].

In recent years, much research has been done on machine recognition of human facial expressions. Feature points ([3]), physical skin and muscle activation models ([21], [33], [30]), optical flow models ([13]), feature based models using manually selected features ([26]), local parametrized optical flow ([4]), deformable contours ([20], [23]), combined with optical flow ([35]) as well as deformable templates ([18],[37],[15],[6]) among several other techniques have been used for facial expression analysis. This paper extends these previous efforts to real-time analysis of the human face using our blob tracking methodology. This extension required development of an incremental Expectation Maximization method, a new mixture-of-Gaussians blob model, and a continuous, real-time HMM classification method suitable for classification of shape data. Applications of this new system, called LAFTER (Lips and Face TrackER) include video-conferencing, real-time computer graphics animation, and "virtual windows" for visualization. Of particular interest is our ability for accurate, real-time classification of the user's mouth shape without constraining head position; this ability makes possible (for the first time) real-time speech-reading and expression recognition in unconstrained office environments.

The paper is structured as follows: the general mathematical framework; use in face detection and tracking; use in mouth detection and mouth tracking; mouth expression recognition; results and applications; conclusions and future work.

2. Mathematical Framework

The notion of grouping atomic parts of a scene together to form blob-like entities based on proximity and visual appearance is a natural one, and has been of interest to visual scientists since the Gestalt psychologists studied grouping criteria early in this century [12].

In modern computer vision processing we seek to group pixels of images together and to "segment" images based on visual coherence, but the "features" obtained from such efforts are usually taken to be the boundaries, or contours, of these regions rather than the regions themselves. In very complex scenes, such as those containing people or natural objects, contour features often prove unreliable and difficult to find and use.

The blob representation that we use was developed by Pentland and Kauth *et al* [25, 19] as a way of extracting an extremely compact, structurally meaningful description of multi-spectral satellite (MSS) imagery. In this method feature vectors at each pixel are formed by adding (x, y) spatial coordinates to the spectral (or textural) components of the imagery. These are then clustered so that image properties such as color and spatial similarity combine to form coherent connected regions, or "blobs," in which all the pixels have similar image properties. This blob description method is, in fact, a special case of recent Minimum Description Length (MDL) algorithms [9, 8, 2].

We have used essentially the same technique for real-time tracking of people in color video [34]. In that application the spatial coordinates are combined with color and brightness channels to form a four-element feature vector at each point $(x, y, \frac{r}{r+g+b}, \frac{g}{r+g+b})$. These were then clustered into blobs to drive a "connected-blob" representation of the person.

By using Expectation Maximization (EM) methods to obtain Gaussian mixture models for the spatio-chrominance feature vector, very complex shapes and color patterns can be adaptively estimated from the image stream. In our system we use an incremental version of EM, which allows us to adaptively and continuously update the spatio-chromatic blob descriptions. Thus not only can we adapt to very different skin colors, etc., but also to changes in illumination.

2.1. Blobs: A Probabilistic Representation

We can represent shapes in both 2-D and 3-D by their low-order statistics. Clusters of 2-D points have 2-D spatial means and covariance matrices, which we shall denote $\overline{q}$ and C_q. The blob spatial statistics are described in terms of their second-order properties; for computational convenience we will interpret this as a Gaussian model. The Gaussian interpretation is not terribly significant, because we also keep a pixel-by-pixel *support map* showing the actual occupancy.

Like other representations used in computer vision and signal analysis, including superquadrics, modal analysis, and eigen-representations, blobs represent the global aspects of the shape and can be augmented with higher-order statistics to attain more detail if the data supports it. The reduction of degrees of freedom from individual pixels to blob parameters is a form of regularization which allows the ill-conditioned problem to be solved in a principled and stable way.

For both 2-D and 3-D blobs, there is a useful physical interpretation of the blob parameters in the image space. The mean represents the geometric center of the blob area (2-D) or volume (3-D). The covariance, being symmetric, can be diagonalized via an eigenvalue decomposition: $C = \Phi L \Phi^T$,where Φ is orthonormal and L is diagonal.

The diagonal L matrix represents the size of the blob along independent orthogonal object-centered axes and Φ is a rotation matrix that brings this object-centered basis in alignment with the coordinate basis of C.

This decomposition and physical interpretation is important for estimation, because the shape L can vary at a different rate than the rotation Φ. The parameters must be separated so they can be treated appropriately.

2.2. Maximum Likelihood Estimation

The blob features are modeled as a mixture of Gaussian distributions in the color (or texture, motion, etc.) space. The algorithm that is generally employed for learning the parameters of such a mixture model is the *Expectation-Maximization (EM)* algorithm of Dempster *et al* [10], [29].

In our system the input data vector d is the normalized R,G,B content of the pixels in the image. The color distribution of each of our blobs is modeled as a mixture of Gaussian Probability Distribution Functions (PDF's) that are iteratively estimated using EM. We can perform a maximum likelihood decision criterium after the clustering is done because human skin forms a dense manifold in *color space*. Two different clustering techniques, both derived from EM are employed: an off-line training process and an on-line adaptive learning process.

In order to determine the mixture parameters of each of the blobs, the unsupervised EM clustering algorithm is computed off-line on hundreds of images of the different classes to be modeled (in our case, face, lips and interior of the mouth), in a similar way as is done for skin color modeling in [17]. When a new frame is available the likelihood of each pixel is computed using the learned mixture model and compared to a likelihood threshold. Only those pixels whose likelihood is above the threshold are classified as belonging to the model.

2.3. Adaptive Modeling via EM

Even though general models make the system relatively user-independent, they are not as good as an adaptive, user-specific model would be. We therefore use adaptive statistical modeling of the blob features to narrow the general model, so that its parameters are closer to the specific users' characteristics.

The first element of our adaptive modeling is to update the model priors as soon as the user's face and face features have been detected. The new distribution parameters are

computed as follows:

$$\begin{aligned}\Sigma_{new} &= [\Sigma^{-1}_{general} + \Sigma^{-1}_{user}]^{-1} \\ \mu_{new} &= \Sigma_{new}[\Sigma^{-1}_{general} * \mu_{general} + \Sigma^{-1}_{user} * \mu_{user}]\end{aligned} \quad (1)$$

Equation 1 corresponds to a *model averaging* of the general and the learned models, in which the prior probability for each model is the same.

This update of priors occurs only at the beginning of the sequence, assuming that the blob features are not going to drastically change during run time. To obtain a fully adaptive system, however, one must also be able to handle second-to-second severe changes in illumination and user characteristics.

We therefore use an *on-line* Expectation-Maximization algorithm ([27]) to adaptively model the image characteristics, We model both the background and the face as a mixture of Gaussian distributions with mixing proportions π_i and K components:

$$p(x/\Theta) = \sum_i^K \pi_i \frac{e^{-1/2(x-\mu_i)^T \Sigma_i^{(-1)}(x-\mu_i)}}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \quad (2)$$

The unknown parameters of such a model are the sufficient statistics of each Gaussian distribution (μ_i, Σ_i), the mixing proportions π_i and the number of components of the mixture K.

The sufficient statistics are updated by computing an on-line version of the traditional EM update rules. If the first n data points have already been computed, the parameters when data point $(n+1)$[1] is read are estimated as follows: First, the *responsibility* h_i^{n+1} for a new data point x^{n+1} is computed:

$$h_i^{n+1} = \frac{\pi_i^n p(x^{n+1}/\theta_i^n)}{\sum_j \pi_j^n p(x^{n+1}/\theta_j^n)} \quad (3)$$

This responsibility can be interpreted as the probability that a data point x^{n+1} was generated by component i. Once this responsibility is known, the sufficient statistics of the mixture components are updated, weighted by the responsibilities:

$$\pi_i^{n+1} = \pi_i^n + \frac{h_i^{n+1} - \pi_i^n}{n} \quad (4)$$

$$\mu_i^{n+1} = \mu_i^n + \frac{h_i^{n+1}}{n * w_i^n}(x^{n+1} - \mu_i^n) \quad (5)$$

$$\sigma_i^{2(n+1)} = \sigma_i^{2(n)} + \frac{h_i^{n+1}}{n * w_i^n}((x^{n+1} - \mu_i^n)^2 - \sigma_i^{2(n)}) \quad (6)$$

where σ_i is the standard deviation of component i and w_i^{n+1} is the *average* responsibility of component i per point: $w_i^{n+1} = w_i^n + \frac{h_i^n - w_i^n}{n}$. The main idea behind this update rules is to distribute the effect of each new observation to all the terms in proportion to their respective likelihoods.

A new component is added to the current mixture model if the most recent observation is not *sufficiently well explained* by the model. If the last observed data point has a very low likelihood with respect of each of the components of the mixture, i.e. if it is an outlier for all the components, then a new component is added with mean the new data point and weight and covariance matrix specified by the user. The threshold in the likelihood can be fixed or could be stochastically chosen: the algorithm would randomly choose whether to add a component or not given an outlier. There is a maximum number of components for a given mixture as well.

The foreground models are initialized with the off-line unsupervised learned *a priori* mixture distributions described above. In this way, the algorithm quickly converges to a mixture model that can be directly related to the *a priori* models' classes. The background models are not initialized with an *a priori* distribution but learned on-line from the image.

2.4. MAP segmentation

Given these models, a MAP foreground-background decision rule is applied to compute *support maps* for each of the classes, that is, pixel-by-pixel maps showing the class membership of each model. Given several statistical blob models that could potentially describe some particular image data, the membership decision is made by searching for the model with the Maximum A Posteriori (MAP) probability.

Once the class memberships have been determined, the statistics of each class are then updated via the EM algorithm, as described above. This approach can easily be seen to be a special case of the MDL segmentation algorithms developed by Darrell and Pentland [9, 8] and later by Ayer and Sawhney [2].

2.5. Kalman filtering

To ensure stability of the MAP segmentation process, the spatial parameters for each blob model are filtered using a zero-order Kalman filter. For each blob we maintain two independent, zero-order filters, one for the position of the blob centroid and another for the dimensions of the blob's bounding box. The MAP segmentation loop now becomes:

1. For each blob predict the filter state vector, $X^* = \hat{X}$ and covariance matrix, $C^* = \hat{C} + (\Delta t)^2 W$, where the matrix W measures the precision tolerance in the estimation of the vector X and depends on the kinematics of the underlying process.

2. For each blob new observations Y (e.g., new estimates of blob centroid and bounding box computed from the

[1] Superscript n will refer in the following to the estimated parameters when n data points have already been processed

image data) are acquired and the Mahalanobis distance between these observations (Y,C) and the predicted state $(\hat{X}, \hat{C})$ is computed. If this distance is below threshold, the filters are updated by taking into account the new observations:

$$\hat{C} = \left[C^{*^{-1}} + C^{-1} \right]^{-1} \quad (7)$$

$$\hat{X} = \hat{C} \left[C^{*^{-1}} X^* + C^{-1} Y \right]^{-1} \quad (8)$$

Otherwise a discontinuity is assumed and the filters are is reinitialized: $\hat{X} = X^*$ and $\hat{C} = C^*$.

A generalized version of this technique is employed in [7] for fusing several concurrent observations.

2.6. Continuous real-time HMMs

Our approach to temporal interpretation of facial expressions uses Hidden Markov Models (HMMs) [28] to recognize different patterns of mouth movement. HMMs are one of the basic probabilistic tools used for time series modeling.

HMMs fall into our Bayesian framework with the addition of time in the feature vector. They offer dynamic time warping, an efficient learning algorithm and clear Bayesian semantics. We have developed a **real-time** HMM system that computes the maximum likelihood of the input sequence with respect to all the models during the testing or recognition phase. This HMM based system runs in real time on an SGI Indy, with the low-level vision processing occurring on a separate Indy, and communications occurring via a socket interface.

3. Automatic Face Detection and Tracking

Our approach to the face finding problem uses coarse color and size/shape information. This approach has advantages over correlation or eigenspace methods, such as speed and rotation invariance under constant illumination conditions. Moreover, our own work [34], or that of Schiele *et al* or Hunke *et al* [31, 16] have shown that use of normalized or chromatic color information $(\frac{r}{r+g+b}, \frac{g}{r+g+b})$ can be reliably used for finding 'flesh areas' present in the scene despite wide variations in lighting.

By training the model on thousands of skin color samples, using off-line EM clustering, we have obtained a model that is valid for a broad spectrum of users (Indian, Asian, Caucasian, South American...).

As described in the mathematical framework (section 2), our system uses an adaptive EM algorithm as well. Both the foreground and background classes are learned incrementally from the data. As a trade-off between the adaptation process and speed, new models are updated only when there is a significant drop in the a posteriori match between model and data.

Two to three mixture components is the typical number required to accurately describe the face. Mouth models are more complex, often requiring up to five components. This is because the mouth model must include not only lips, but also the interior of the mouth and the teeth.

3.1. Blob Growing

After initial application of the MAP decision criterion to the image, often isolated and spurious pixels are misclassified. Thus local pixel information needs to be merged into connected regions that correspond to each of the blobs.

The transition from local to global information is achieved by applying a connexity algorithm which grows the blob. The algorithm we use is an speed-optimized version of a traditional connectivity algorithm that considers for each pixel the values within a neighborhood of a certain radius (which can be varied at run-time) in order to determine whether this pixel belongs to the same connected region.

Finally, these blobs are then filtered to obtain the best candidate for being a face or a mouth. Color information alone is not robust enough for this purpose. The background, for instance, may contain skin colors that could be grown and erroneously considered as faces. Additional information is thus required. In the current system, geometric information, such as the size and shape of the object to be detected (faces) is combined with the color information to finally locate the face. In consequence, only those 'skin blobs' whose size and shape (ratio of aspect of its bounding box) are closest to the canonical face size and shape are considered. The result is shown in figure 1.

Figure 1. Face detection and growing

3.2. Active Camera Control

Because our system already maintains a Kalman filter estimate of the centroid and bounding box of each blob, it is a relatively simple matter to use these estimates to control the camera so that the face of the user always appears in the center of the image and with the desired size. Our system uses an abstraction of the camera control parameters, so that different camera/motor systems (currently the Canon VCC1 and Sony EVI-D30) can be successfully used in a totally transparent way. In order to increase tracking performance, the camera pan-tilt-zoom control is done by an independent light-weight process (thread) which is started by the main program.

The current estimation of the position and size of the user's face provides a reference signal to a PD controller

which determines the tilt, pan and zoom of the camera so that the target (face) has the desired size and is at the desired location. The zoom control is relatively simple, because it just has to be increased or decreased until the face reaches the desired size. Pan and tilt speeds are controlled by $S_c = \frac{C_e * E + C_d * \frac{dE}{dt}}{F_z}$, where C_e and C_d are constants, E is the error, i.e. the distance between the face current position and the center of the image, F_z is the zoom factor, and S_c is the final speed transmitted to the camera.

The zoom factor plays a fundamental role in the camera control because the speed with which the camera needs to be adjusted depends on the displacement that a fixed point in the image undergoes for a given rotation angle, which is directly related to the current zoom factor. The relation between this zoom factor and the current camera zoom position follows a non-linear law which needs to be approximated. In our case, a second order polynomial provides a good approximation.

4. Mouth Extraction and Tracking

Once the face location and shape parameters are known (center of the face, width, height and image rotation angle), we can use anthropometric statistics to define a bounding box within which the mouth must be located.

The mouth is modeled using the same principles as the face, i.e. through a second-order mixture model that describes both its chromatic color and spatial distribution. However to obtain good performance we must also produce a more finely detailed model of the face region surrounding the mouth. The face model that is adequate for detection and tracking is not adequate for accurate mouth shape extraction.

Our system, therefore, acquires image patches from around the located mouth and builds a Gaussian mixture model. In the current implementation, skin samples of three different facial regions around the mouth are extracted during the initialization phase and their statistics are computed. Figure 2 is an example of how the system performs in the case of facial hair. The robustness of the system is increased

Figure 2. Head and mouth tracking: rotations, facial hair

by computing at each time step the linearly predicted position of the center of the mouth. A confidence level on the prediction is also computed, depending on the prediction error. When the prediction is not available or its confidence level drops below a threshold, the mouth's position is reinitialized.

4.1. Mouth shape

The mouth shape is characterized by its area, its spatial eigenvalues (e.g., width and height) and its bounding box. The use of this feature vector to classify facial expressions has been suggested by psychological experiments [36, 22], which examined the most important discriminative features for expression classification.

Rotation invariance is achieved by computing the face's image-plane rotation angle and rotating the region of interest with the negative of this angle. Therefore even though the user might turn the head the mouth always appears nearly horizontal, as figure 2 illustrates.

5. Speed, Accuracy, and Robustness

Running LAFTER on a single SGI Indy with a 200Mhz R4400 processor, the average frame rate for tracking is typically 25 Hz. When mouth detection and parameter extraction are added to the face tracking, the average frame rate is 14 Hz.

To measure LAFTER's 3D accuracy during head motion, the RMS error was measured by having users make large cyclic motions along the X, Y, and Z axes respectively, with the true 3D position of the face being determined by manual triangulation. In this experiment the camera actively tracked the face position, with the image-processing/camera-control loop running at a nearly constant 18hz. The image size was 1/6 full resolution, i.e. 106x80 pixels, and the camera control law varied pan, tilt, and zoom to place the face in the center of the image at a fixed pixel resolution.

The RMS error between the true 3D location and the system's output was computed in pixels and is shown in table 1. Also shown is the variation in apparent head size, e.g., the system's error at stabilizing the face image size. As can be seen, the system gave quite accurate estimates of 3D position. Perhaps most important, however, is the robustness

	X RMS (pixels)	Y RMS (pixels)	Translation Range (cm)
Static Face	0.5247 (0.495 %)	0.5247 (0.6559 %)	0.0
X translation	0.6127 (0.578 %)	0.8397 (1.0496 %)	$\pm$ 76
Y translation	0.8034 . (1.0042 %)	1.4287 (1.7859 %)	$\pm$ 28
Z translation	0.6807 (0.6422 %)	1.1623 (1.4529 %)	$\pm$ 78
	Width Std (pixels)	Height Std (pixels)	Size change (pixels)
Zooming	2.2206 (2.09 %)	2.6920 (3.36 %)	Max. size: 86x88 Min. size: 14x20

Table 1. Accuracy.

of the system. LAFTER has been tested on hundreds of

users at many different events, each with its own lighting and environmental conditions. Examples are the *Digital Bayou*, part of SIGGRAPH 96', the *Second International Face & Gesture Workshop (October 96)* or the last three sponsors open houses (1996-1997).

6. Recognition

Using the mouth shape feature vector described above, we trained 5 different HMM's for each of the following mouth configurations (illustrated in figure 3): neutral or default mouth position, extended/smile mouth, sad mouth, open mouth and extended+open mouth (such as in laughing). The neutral mouth acted to separate the various expressions,

Figure 3. Smile-open, sad, open and smile

much as a silence model acts in speech recognition. The final HMM's we derived for the non-neutral mouth configurations consisted of 4-state forward HMM's. The neutral mouth was modeled by a 3-state forward HMM.

Recognition results for a eight different users making over 2000 expressions are summarized in table 2. The users were divided in different groups for training and testing purposes. The first of the recognition tasks shown in table 2 corresponds to a training and testing with all the eight users. The total number of examples is denoted by N, having a total N=2058 instances of the mouth expressions (N=750 for training and N=1308 for testing). As can be seen, accurate classification was achieved in each case.

Recognition Results	On training	On testing
All users	97.73	95.95
Single user	100.00	100.00

Table 2. Recognition results: training and testing data

7. Applications

7.1. Automatic Camera Man.

The static nature of current video communication systems induces extra articulatory tasks that interfere with real world activity. For example, users must keep their head (or an object of interest) within the field of the camera (or of the microphone) in order to be perceived by distant parties. As a result, the user ends up being more attentive to the way how to using the interface than to the conversation itself. The communication is therefore degraded instead of enriched.

In this sense, LAFTER, with its active camera face tracking acts as an 'automatic camera man' that is continuously looking at the user while he/she moves around or gestures in a video-conference session. In informal teleconferencing testing, users have confirmed that this capability significantly improves the usability of the teleconferencing system.

We can also use the system in a 'Virtual Window' mode [14]: as the user moves in front of his local camera, the distant motorized camera is moved in the same way. In informal tests, users have said that the LAFTER-based virtual window system gives a good sense of the distant space.

7.2. Real-time computer graphics animation

Because LAFTER continuously tracks face location, image-plane face rotation angle, and mouth shape, it is a simple matter to use this information to obtain real-time animation of a computer graphics character. This character can, in a simpler version, constantly mimic what the user does (as if it where a virtual mirror) or, in a more complex system, understand (recognize) what the user is doing and react to it. A 'virtual mirror' version of this system — using the character named Waldorf shown in figure 4 — was exhibited in the Digital Bayou section of SIGGRAPH'96 in New Orleans' naive users.

Figure 4. Real time computer graphics animation

7.3. Preferential Coding

Finally, LAFTER can be used as the front-end to a *preferential image coding system*. It is well-known that people are most sensitive to coding errors in facial features. Thus it makes sense to use a more accurate (and more expensive) coding algorithm for the facial features, and a less accurate (and cheaper) algorithm for the remaining image data [11, 24, 1]. Because the location of these features is detected by our system, we can make use of this coding scheme.

8. Conclusion and Future Work

In this paper we have described a real-time system for finding and tracking a human face and mouth, and recognizing mouth expressions using HMM's. The system runs on a single SGI Indy computer, and produces estimates of head position that are surprisingly accurate.

The system has been successfully used as the base for several different applications, including an automatic camera man, a virtual window video communications system,

and a real-time computer graphics animation system. The system has been tested on hundreds of naive users in several physical locations, with a success rate of over 97%.

References

[1] K. Aizawa and T. Huang. Model-based image-coding: Advanced video coding techniques for very-low bit-rate applications. *PIEEE*, 83(2):259–271, February 1995.

[2] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV95*, pages 777–784, 1995.

[3] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *PAMI*, 15(6):602–605, June 1993.

[4] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. volume 1, pages 374–381. IEEE, 1995.

[5] A. Bobick and R. Bolles. The representation space paradigm of concurrent evolving object descriptions. *PAMI*, 14(2):146–156, February 1992.

[6] C. Bregler and S. Omohundro. *Advances in Neural Information Processing Systems*, chapter Surface Learning with Applications to Lipreading. Morgan Kaufmann, San Francisco, CA, 1994.

[7] J. Crowley and F. Bérard. Multi-modal tracking for video-communication. Puerto Rico, June 1997. CVPR.

[8] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *PAMI*, 17(5):474–487, May 1995.

[9] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by minimal description. In *ICCV90*, pages 173–177, 1990.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via de em algorithm. *Journal of the Royal Statistical Society*, 39-B:1–38, 1977.

[11] A. Eleftheriadis and A. Jacquin. Model-assisted coding of video teleconferencing sequences at low bit rates. In *ISCAS*, May-June 1994.

[12] W. Ellis. A source book of gestalt psychology. In *Harcourt, Brace and Co.*, 1939.

[13] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *ICCV95*, pages 360–367, 1995.

[14] W. Gaver, G. Smets, and K. Overbeeke. A virtual window on media space. CHI, 1995.

[15] H. Hennecke, K. Venkatesh, and D. Stork. Using deformable templates to infer visual speech dynamics. Technical Report 9430, California Research Center, June 1994.

[16] H. Hunke. Locating and tracking of human faces with neural networks. Technical report, CMU, Pittsburgh PA, August 1994.

[17] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. Technical Report 401, MIT Media Lab, Cambridge MA, 1996. To appear in Proceedings of IEEE CVPR'97.

[18] M. Kass, A. Witkin, and D. Terzopolous. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, January 1988.

[19] R. Kauth, A. Pentland, and G. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of mss imagery. In *11th Int'l Symp. on Remote Sensing of the Environment*, Ann Harbor MI, 1977.

[20] R. Magnolfi and P. Nesi. Analysis and synthesis of facial motions. volume 1, pages 308–313, Zurich, 1995. International Workshop on Automatic Face and Gesture Recognition.

[21] K. Matsuno, C. Lee, S. Kimura, and S. Tsuji. Automatic recognition of human facial expressions. In *CVPR*, volume 1, pages 352–359. IEEE, 1995.

[22] S. Morishima. Emotion model. pages 284–289, Zurich, 1995. International Workshop on Automatic Face and Gesture Recognition.

[23] Y. Moses, D. Reynard, and A. Blake. Determining facial expressions in real time. volume 1, pages 332–337, Zurich, 1995. International Workshop on Automatic Face and Gesture Recognition.

[24] K. Ohzeki, T. Saito, M. Kaneko, and H. Harashima. Interactive model-based coding of facial image sequence with a new motion detection algorithm. *IEICE*, E79B(10):1474–1483, October 1996.

[25] A. Pentland. Classification by clustering. In *IEEE Symp. on Machine Processing and Remotely Sensed Data*, Purdue, IN, 1976.

[26] I. Pilowsky, M. Thornton, and B. Stokes. *Aspects of face processing*, chapter Towards the quantification of facial expressions with the use of a mathematics model of the face, pages 340–348.

[27] C. Priebe. Adaptive mixtures. *Journal of the American Statistical Association*, 89(427), 1994.

[28] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[29] R. Redner and H. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26:195–239, 1984.

[30] M. Rydfalk. *CANDIDE: A parametrized face*. PhD thesis, Linköpnig University, EE Depart., Oct 1987.

[31] B. Schiele and A. Waibel. Gaze tracking based on face color. In *International Workshop on Automatic Face and Gesture Recognition*, pages 344–349, 1995.

[32] T. Starner and A. Pentland. Real-time asl recognition from video using hmm's. Technical Report 375, MIT, Media Laboratory, MIT, Media Laboratory, Cambridge, MA 02139, 1996.

[33] K. Waters. A muscle model for animating three-dimensional facial expression. volume 21(4), pages 17–23. ACM SIGGRAPH Conf. Proceedings, 1987.

[34] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *Photonics East, SPIE*, volume 2615, Bellingham, WA, 1995.

[35] Y. Yacoob and L. Davis. Recognizing human facial expressions from long image sequences using optical-flow. *PAMI*, 18(6):636–642, June 1996.

[36] H. Yamada. Dimensions of visual information for categorizing facial expressions of emotion. *Japanese Psychological Research*, 35(4), 1993.

[37] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *Journal of Computer Vision*, pages 99–111, 1992.