Learning Low-Complexity Face Models for Tracking and Recognition

A Proposal for a Doctoral Dissertation

by

Zakaria Ajmal

Ph.D. Dissertation Advisor
Dr. Russell M. Mersereau

School of Electrical and Computer Engineering
Georgia Institute of Technology
February 2003

Contents

List of Tables

List of Figures

1 Introduction

The human face has been an important subject of research in the computer vision community for quite some time. People use faces to communicate in various ways. Lip-reading helps with understanding speech, while facial expressions provide emotional cues. Researchers have worked on compression of video sequences of facial images for videophone-like applications; on creating a virtual presence for video conferencing; and on the analysis and synthesis of facial expressions and speech for human-computer interaction.

For these applications, a number of researchers have taken a model-based approach. They have developed a model of the human face and then used the deformation modes of the model to track the face in a video sequence. In this proposal, we outline some of these techniques for developing human facial models as well as tracking faces using those models. We concentrate on mesh models instead of muscle models for the face as they are simpler and computationally faster.

The following is a summary of the major contributions of this proposal:

1. A better method to automatically learn a simple 3-D face model from training stereo image sequences.

2. A novel algorithm to compute functional subspaces of the face, expression and identity, simultaneously from training data [1].

3. A contribution to tracking the 3-D pose, position and facial expression in a monocular image sequence using our 3-D model.

4. Using the low-dimensional identity subspace of our model to recognize faces.

5. Recognizing facial expressions using the action subspace of our model.

We start with a review of related work in face modelling using 2-D and 3-D meshes and model-based tracking of faces. We then present the thesis work that has been completed and outline the work that remains to be done.

2 Previous Work

There has been a large body of research on the tracking, recognition and modelling of human faces [2, 3, 4, 5, 6]. We will describe some of the important work related to this proposal below.

2.1 Manually Constructed Face Models

Most researchers doing model-based face tracking have manually constructed face models using anthropometric data. DeCarlo and Metaxas [7] used a deformable model that consists of several articulated points and has to be designed by hand prior to tracking. They model both shape and expression. The facial expression part of their model is based on Facial Action Coding System (FACS) [8] and only a few facial expressions are modelled. Because of this limitation, they cannot accurately track other expressions, for example lip puckering and speech.

Eisert and Girod [9] use a 3-D face model based on the popular wireframe CANDIDE [10] shown in Figure 1. To decrease the number of explicit control points, they construct the head surface using B-splines [11]. The shape and texture of the model is adapted to an individual using a 3-D laser scan. They track facial motion using MPEG-4 Synthetic Natural Hybrid Coding (SNHC) Facial Animation Parameters (FAP) [12]. Since their application is video coding, they are mostly interested in an accurate reconstruction of the face image. Using a hierarchical optical flow

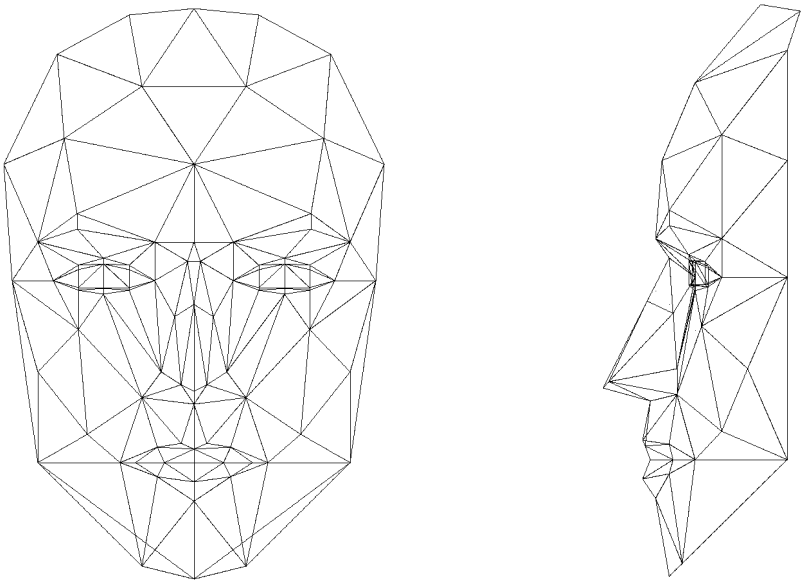


Figure 1: CANDIDE-3: Face model

based method, analysis-by-synthesis feedback, and estimation of photometric effects, they can achieve bit rates as low as 1 kbits/sec at 34 dB PSNR [13] for the motion parameters. Estimation of the illumination also improves motion parameter accuracy.

Ahlberg [6] also uses a manually constructed CANDIDE-3 (See Figure 1) [14]. He [15] extends the directed search algorithm developed by Cootes et al [5] for the Active Appearance Model to work with his parameterized face model. The model itself is hand-crafted, but the gradient matrix for the tracking is trained on facial images. The CANDIDE-3 models texture, shape and action and has been designed to be compatible with MPEG-4 facial definition points (FDP) and facial animation parameters (FAP).

2.2 Shape/Texture Models Learned from Data

Principal Component Analysis (PCA) is a powerful method for estimating a low-dimensional representation of data from a higher-dimensional space. Bregler et al. [16] and Blanz and Vetter [17] have used PCA to approximate non-rigid shapes as linear combinations of sets of rigid basis shapes.

Cootes and Taylor [18] learn an active appearance model (AAM) from training data using PCA. The AAM consists of a shape (geometry) model and an appearance (texture) model. The appearance model is computed after warping the image to a generic shape. The model is learned using training images with landmarks. In addition to the shape and appearance basis vectors, the gradient matrix for small changes in the model parameters is also learned. When fitting a model to an image, the model parameters are updated using the gradient matrix based on the error between the image and the synthesized model. Cootes et al [19] model all the variations in the data together instead of modelling the shape, action and illumination separately. Also, since their models of the face are two-dimensional, they have to use view-based appearance models to generate different poses.

Gokturk et al [20] have developed a user-specific face model, which is the precursor to our current work [1]. In their system, facial expressions are modelled by a linear combination of basis vectors computed using Principal Component Analysis. The differences in face geometry among people are accounted for by individual neutral face shapes.

Costen et al [21, 22, 23] use a 2-D active appearance model to estimate the functional subspaces: lighting, pose, identity and expression. Separate ensembles of images with variations in a particular subspace are used. Initial estimates of the

subspaces are computed from PCA of these ensembles. Since the ensembles are somewhat contaminated with contribution from other subspaces, an iterative algorithm is then applied to the coded image to maximize the probability of coding across these subspaces. The subspace ensembles are recalculated by a weighted projection on each subspace. This algorithm then separates the different subspaces. The performance of an face recognition system on the resulting identity subspace is better than on the original AAM data.

2.3 Optical Flow Tracking

Lucas-Kanade algorithm is an image alignment algorithm using optical flow which was first described in a seminal paper by Lucas and Kanade [24].

DeCarlo and Metaxas [4, 7] combine optical flow and forces computed from edges for a 3-D deformable face model to estimate facial motion. The optical flow constraint equations are transformed into a system which constrains the velocities of the motion parameters of the model. They use an iterated extended Kalman filter to relax the optical flow constraint as noise increases to make their system more robust [25]. Instead of fixing the shape parameters after the first frame, they [26] make minor adjustments using the optical flow residuals so that the motion error is minimized.

2.4 Face Recognition Using Models

Face recognition has been an active area of research for quite some time. Eigenfaces [27] are one of the common methods for face recognition. Another approach is to use facial features, like the distance between the eyes, etc.

Edwards et al [28] use 2-D AAM to recognize faces. They use linear discriminant

analysis to separate the identity information from pose, expression and lighting, etc.

Blanz et al [29, 30, 31] use a 3-D morphable model to recognize faces across variations in pose and illumination. Their algorithm estimates the 3-D shape and texture of faces from a single image by fitting a statistical, morphable model. The deformable model is learned from a set of textured 3-D scans of heads. However, the algorithm requires manual initialization of 6–8 feature points on the face image.

2.5 Expression Recognition Using Models

There has been a lot of research to recognize facial expressions. Most of the research has focussed on 2-D images and local features. According to a survey [32], the systems that implement a combination of holistic and local image analysis and classify facial expressions with the FACS coding scheme give the best results.

Lanitis et al [33] were able to get a 70% recognition accuracy, on a database of 139 training images and 118 test images with 30 subjects, for seven facial expressions using the active appearance model. They trained a classifier on both the shape and appearance parameters of training images.

Gokturk et al [34] classified facial expressions using a support vector machine (SVM) on the action parameters estimated from model-based tracking.

3 Completed Work

In this section, we summarize the completed work that is the foundation of this proposal.

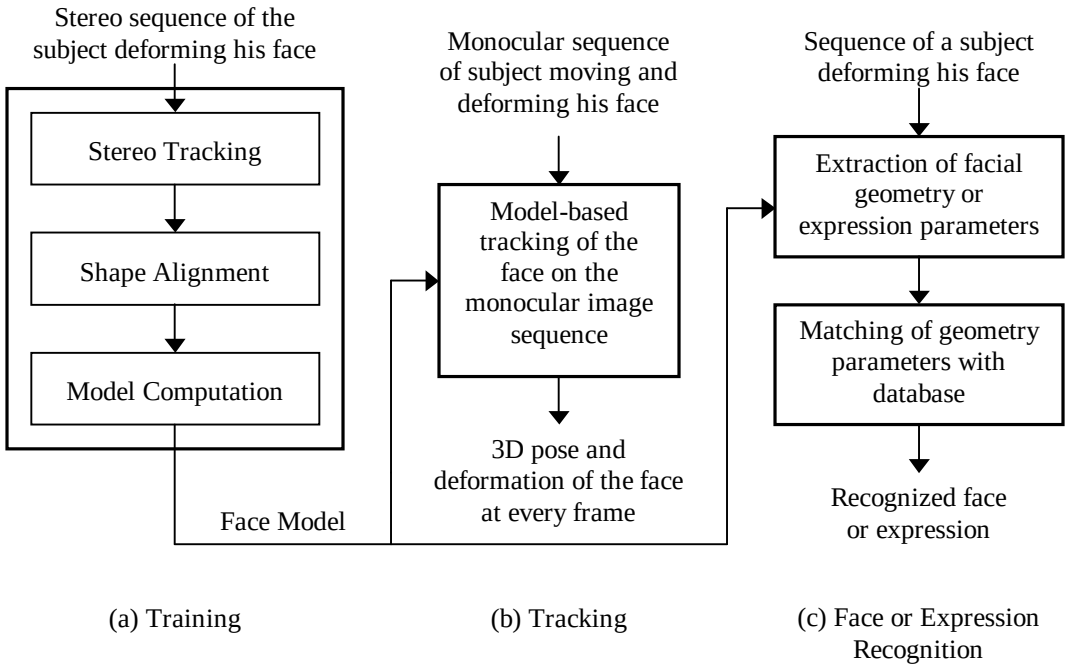


Figure 2: Main components of the algorithm

3.1 Description of the System

Figure 2 shows the main components of our system. There are three distinct components. The main component of our system is the learning phase as shown in Figure 2(a). Our system models the identity of the subject (shape) and facial expressions (action). In the learning stage, the training data, consisting of stereo image sequences, is used to learn the mean shape and the modes of variation for shape and action of the model. For this purpose, a low-complexity mesh is initialized on the face of a subject in the training sequences and then tracked through the sequence using optical flow techniques. The resulting sequence of meshes for different subjects are aligned and used to compute a compact deformable model of the face using Principal Component Analysis. This model can then be fitted to a specific subject by varying the coefficients of the variation modes of the model.

The other two components of the system show some of the applications for our face model. In Figure 2(b), the learned face model is used for monocular tracking. A subject is free to change his pose, position and facial expression. The system uses a model-based optical flow technique to track the action parameters (facial expression), and orientation and position in three dimensions.

Figure 2(c) shows the use of our face model for recognizing faces and expressions. Since we explicitly model the geometric shape and facial expressions, we can use the shape parameters to recognize faces and action parameters to recognize facial expressions. Some initial work for these applications will be discussed in this proposal.

We use real data to construct a face model instead of manually constructing one because it allows us to capture the shape and action deformation spaces with a small, optimum number of basis vectors. Also, it would allow automatic enrollment of subject into the model. We can check the generalization properties of the model by the closeness of the fit of the model and the tracking performance on test subjects not included in the training set.

3.2 Face Model

We model the face using a 3-D mesh consisting of N points. We are using a mesh model of $N = 19$ points, with nodes located on the eyes (2), nose (3), mouth (8) and eyebrows (6) (See Figure 3).

The face reference frame is attached to the head. Therefore, at frame n, the coordinates of the ith node on the face mesh in the camera reference frame, $\mathbf{X}_c^i(n)$, are related to the coordinates in the face reference frame, $\mathbf{X}^i(n)$, by a rigid transformation.

$$\mathbf{X}_c^i(n) = \mathbf{R}(n)\mathbf{X}^i(n) + \mathbf{t}(n) \qquad \text{for } i = 1, \ldots, N \tag{1}$$

(a) Left image (b) Right image

Figure 3: Face mesh shown for one frame

where $\mathbf{R}(n)$ and $\mathbf{t}(n)$ are the rotation matrix and translation vector for frame n. $\mathbf{R}(n)$ is uniquely parameterized by a rotation vector $\omega(n)$. The rotation vector and the rotation matrix are related by the Rodrigues formula [35].

Let $\mathbf{x}^i(n)$ be the 2-D image coordinates of the point $\mathbf{X}^i(n)$. The projection of the 3-D point onto the image is a function of the camera model. For a traditional pinhole camera model

$$\mathbf{x}^i(n) = \begin{bmatrix} x_n^i(n) \\ y_n^i(n) \end{bmatrix} = \begin{bmatrix} X_c^i(n)/Z_c^i(n) \\ Y_c^i(n)/Z_c^i(n) \end{bmatrix} = \pi_c(\mathbf{X}^i(n), \omega(n), \mathbf{t}(n)) \tag{2}$$

The face is modelled as the sum of a mean shape $\overline{\mathbf{X}}^i$ and a linear combination of shape (identity) and action (expression) basis vectors.

$$\mathbf{X}^i(n) = \overline{\mathbf{X}}^i + \mathbf{S}^i\sigma(n) + \mathbf{A}^i\alpha(n) \tag{3}$$

where $\mathbf{S}^i$ and $\mathbf{A}^i$ are matrices of size $3 \times p$ and $3 \times q$ respectively (p and q are the respective dimensions of the shape and action spaces.) The columns of $\mathbf{S}^i$ and $\mathbf{A}^i$ are the shape and action basis vectors respectively, and $\sigma(n)$ and $\alpha(n)$ are the shape and action parameter vectors for frame n.

In the training phase, the mean shape $\overline{\mathbf{X}}^i$ and the shape and action basis vectors $\mathbf{S}$ and $\mathbf{A}$ are learned from the stereo training data.

During tracking, the pose parameters, i.e., rotation vector $\omega(n)$ and translation vector $\mathbf{t}(n)$, and the deformation parameters $\sigma(n)$ and $\alpha(n)$ are estimated from the image sequence under the model constraints of equations 1, 2 and 3.

3.3 Stereo Tracking of Training Data

To train the model, we need the 3-D face mesh for a variety of subjects and expressions. The stereo image sequences provide us with the subjects making a variety of facial expressions. Unlike previous researcher, we use stereo tracking to get the 3-D geometry data instead of marking the points manually on the images. Since we have a 3-D model, we do not need to explicitly model for pose variation (unlike 2-D AAM). Hence, to keep the tracking simpler, the subjects do not change their pose while making different facial expressions.

The 19-node mesh on the first frame is initialized semi-automatically. (See Figure 3). These points are then tracked using 3-D optical flow techniques [24, 36] constrained by space-energy functions similar to those described in [20].

A nodal point $P^i = \mathbf{X}_c^i(n)$ on the mesh is tracked from frame $n-1$ to frame n by minimizing the following cost function

$$E_i(n) = \sum_{ROI} \left\{ \begin{array}{rl} & (1-\gamma)[I_n^L(\mathbf{x}_L^i(n)) - I_{n-1}^L(\mathbf{x}_L^i(n-1))]^2 \\ + & (1-\gamma)[I_n^R(\mathbf{x}_R^i(n)) - I_{n-1}^R(\mathbf{x}_R^i(n-1))]^2 \\ + & \gamma[I_n^R(\mathbf{x}_R^i(n)) - I_{n-1}^L(\mathbf{x}_L^i(n-1))]^2 \\ + & \gamma[I_n^L(\mathbf{x}_L^i(n)) - I_{n-1}^R(\mathbf{x}_R^i(n-1))]^2 \end{array} \right\} \tag{4}$$

where I_n^L and I_n^R are the left and right images at frame n and $\mathbf{x}_L^i(n)$ and $\mathbf{x}_R^i(n)$ are the

coordinates of the left and right image projections of P^i. The tracking using this cost function has drift problems. Therefore we added a few regularization terms so that the 3-D mesh preserves its integrity while deforming smoothly. These regularization terms are similar to those used by Gokturk et al [20] and DeCarlo and Metaxas [4, 7, 25, 26].

$$E(n) = \left\{ \begin{array}{ll} & \sum_i E_i(n) \\ + & \sum_i \rho_i \|d\mathbf{X}_c^i(n)\|^2 \\ + & \sum_{i,j} \beta_{ij} \|d\mathbf{X}_c^i(n) - d\mathbf{X}_c^j(n)\|^2 \\ + & \sum_{i,j} \delta_{ij} (\|\mathbf{X}_c^i(n) - \mathbf{X}_c^j(n)\|^2 - \|\mathbf{X}_c^i(1) - \mathbf{X}_c^j(1)\|^2)^2 \end{array} \right\} \tag{5}$$

where $d\mathbf{X}_c^i(n) = \mathbf{X}_c^i(n) - \mathbf{X}_c^i(n-1)$ and the coefficients ρ_i, β_{ij} and δ_{ij} have different values for different nodes and edges. The edges P^iP^j which are very nonrigid (i.e., subject to large stretches), for example the edges corresponding to lower lip nodes, are assigned smaller values of β_{ij} and δ_{ij}. Similarly, nodes on a more rigid portion of the face are assigned a higher value of ρ_i.

The quality of the tracking was judged visually based on the reprojection of the mesh points on the left and right camera images. The maximum reprojection error of the mesh onto the left and right images was estimated to be less than 2 pixels.

Even though the subjects were asked not to move their heads while recording these sequences, there is still some small movement and rotation. Also, different subjects had their faces at somewhat different locations with respect to the camera. Therefore, the tracked 3-D meshes for each subject have to be aligned to each other. A popular method of aligning shapes is Procrustes analysis [37]. This method applies a rigid transformation (rotation and translation) to each mesh to minimize the sum of distances of each shape to the mean. We use Procrustes analysis to first align the meshes for each subject separately to the mesh corresponding to that subject's neutral face. Then all the subjects are aligned together using the same method.

Points that are known to be highly deformable (for example, on the mouth) are weighed less in the transformation. In our experiments, we set the weight to 0.9 for the eyes and nose and 0.1 for the lips and eyebrows. According to our observation, the alignment and shape learning is not very sensitive to these weights.

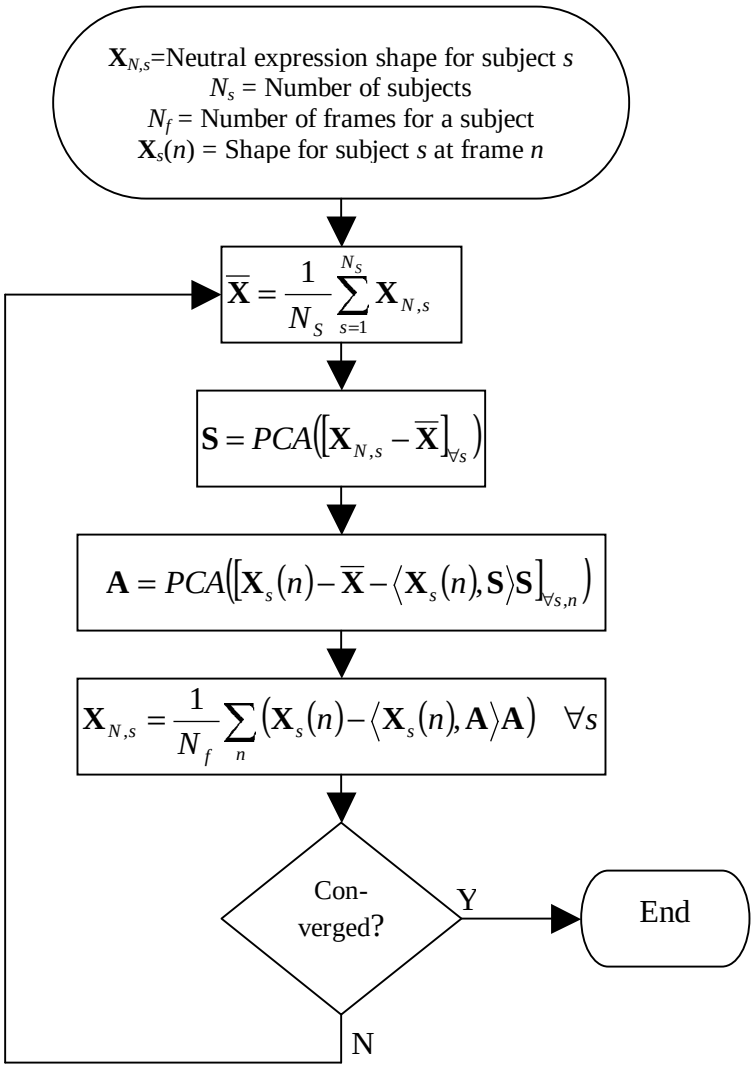
3.4 Model Learning from Data

We want to compute a general face model from the tracking of the training sequences. As mentioned in section 3.2, we need to calculate the model mean shape $\overline{\mathbf{X}}^i$, the shape matrix $\mathbf{S}^i$ corresponding to the identity of the subject and the action matrix $\mathbf{A}^i$ corresponding to facial expressions. We use Principal Component Analysis for the computation.

Referring to Figure 4, in the first iteration, the mean shape is computed as the mean of the neutral expression meshes $\mathbf{X}_{N,s}$ of all the subjects $1, \ldots, N_S$. Equal weighting is given to each subject.

Next, the mean shape is subtracted from the neutral expression shapes of all the subjects. The results of this operation are stacked as columns in a matrix and principal components of this matrix are calculated. We found that the first $p = 2$ principal components could explain most of the variance in the data and hence kept these two components as the shape basis vectors in the matrix $\mathbf{S}$.

Then, taking all the face meshes, we subtract off the mean shape and the contribution of the shape basis vectors. The resulting residuals should ideally consist of only contributions from the facial expressions. We assemble this result in a matrix and use Principal Component Analysis to compute the action basis vectors from this matrix. The most significant $q = 5$ vectors are kept in the action matrix $\mathbf{A}$.



$\overline{\mathbf{X}}$ = Mean shape

$\mathbf{S}$ = Matrix of Shape Basis Vectors

$\mathbf{A}$ = Matrix of Action Basis Vectors

$\mathbf{X}_{N,s}$ = Estimate of neutral face shape for subject s

Figure 4: Flowchart for the learning of the face model

Since the neutral expression meshes are not ideal, the shape basis vectors do have some component of facial expression. To remove that component, we recompute estimates of the neutral meshes from all the mesh sequences. We subtract off the projection of the action vectors from the mesh data. Having removed the action component from the meshes, the results are averaged together for each subject giving us an estimate of the neutral expression meshes.

We now iterate again using this estimate of the neutral expression shapes to re-estimate the mean shape and the shape matrix as described earlier. The action matrix is then recalculated and the process is iterated until the solution converges. The algorithm converges in about 10–20 iterations.

To check robustness, we excluded one subject at a time from the training set and computed the model in each case. The similarity between the shape and action subspace estimates when all the eleven subjects were included in the training and the subspace estimates when one subject was excluded gives us an idea of the robustness of our algorithm. We measured similarity between the subspaces by projecting one subspace onto the other and calculating the error between the original subspace and its projection. We calculated the similarity measure between the "all subjects in training set" model and each of the models where one subject was excluded. The average similarity measures for the shape matrix $\mathbf{S}$ and the action matrix $\mathbf{A}$ were 0.988 and 0.948 (where 1 means exactly the same) respectively while the mean shape $\overline{\mathbf{X}}$ varied by an average of 1.7% [1].

In another experiment, we computed the model approximations for the neutral face shapes for each subject when that subject is not included in the training database. The rms error between the nodes of the model and the actual mesh nodes was 3.53mm. Figure 6 shows the projection of the models on the left and right images of two subjects

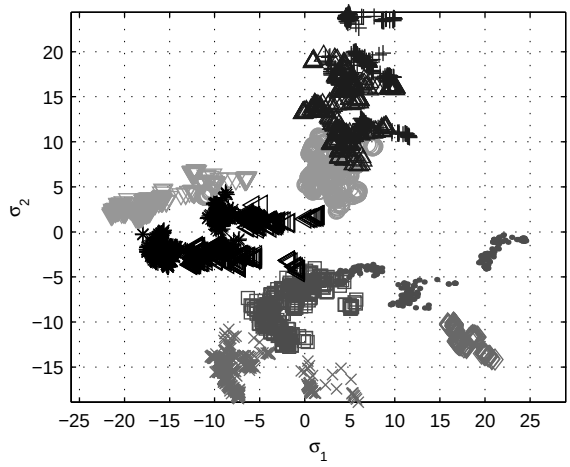


Figure 5: Projections of all the meshes in σ space

(circles) and the actual mesh points (shown as plus signs).

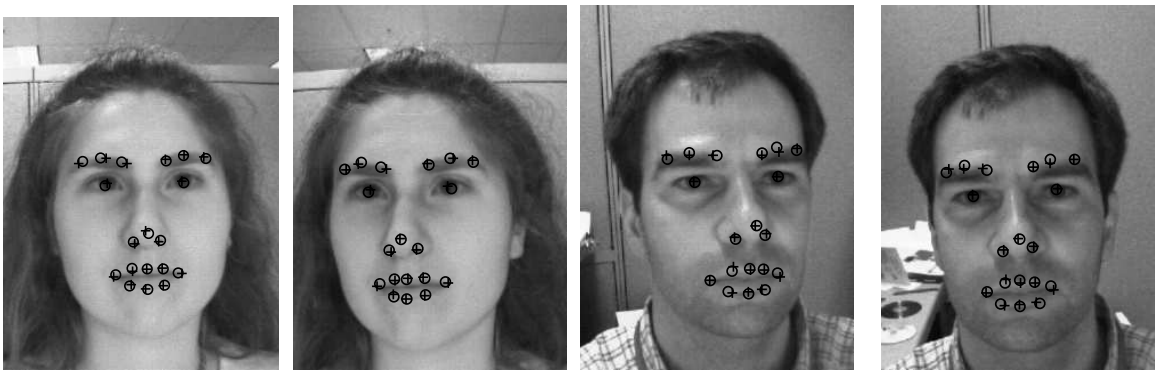


Figure 6: Model estimates for two subjects. The + is for model approximation and the circle for actual mesh points.

To find out if the shape (identity) space in the computed model actually covers the identities of the subjects in our training set, we estimated the 2-dimensional shape parameter vector σ for all the frames of the training image sequences. The plot in σ-space is shown in Figure 5. It can be seen that each subject's shape parameters are somewhat clustered together. This shows that the shape vector represents the differences in shape between faces and hence provides information about the identity

of the subjects.

3.5 Tracking Monocular Sequences Using Model

Optical flow tracking computes the translational displacement of points in the image given two successive frames [24, 36]. Each point in the image is processed independently. In model-based tracking, we constrain the points using a parameterized model (see equation 3). Hence, we estimate the model parameters simultaneously from image measurements, similar to Gokturk et al [20].

Since the shape parameter vector σ defines the identity of a subject, it is initialized at the start of the sequence and is kept constant over the sequence.

Assuming that the face model has been tracked from the first frame to the $(n-1)$th frame I_{n-1}, our objective is to estimate the optimal pose (rotation vector $\omega(n)$ and translation vector $\mathbf{t}(n)$) and the action parameter vector, $\alpha(n)$, that best fit the subsequent frame I_n. The cost function to be minimized for the purpose is given by

$$C_n = \sum_{i,ROI} \Big\{ (1-\epsilon)(I_n(\mathbf{x}^i(n)) - I_{n-1}(\mathbf{x}^i(n-1)))^2 + \epsilon(I_n(\mathbf{x}^i(n)) - I_1(\mathbf{x}^i(1)))^2 \Big\} \quad (6)$$

The first term in equation 6 is the standard matching cost used in the Shi-Tomasi-Kanade tracker [24, 36]. The second term measures the image mismatch between the current image I_n and the first image I_1. This additional term weakly forces the facial features to appear the same over the complete sequence. Hence, it avoids tracking drift and increases robustness. It is referred to as the drift monitoring term.

In our experiments, the initialization of the face mesh in the first frame is done semi-automatically. The value of the drift monitoring coefficient was set to $\epsilon = 0.2$ for all our experiments to emphasize standard tracking cost over drift monitoring cost.

To test our face model and tracking algorithm, we used some test sequences. The subjects in these test sequences were not included in the training database. In these sequences, the subject rotates and moves his/her head covering a working volume of 15cm $\times$ 15cm $\times$ 15cm while doing a variety of facial expressions. Figures 7 and 8 show some of the results from the tracking of these sequences. It can be seen that the tracking of the face is maintained over the length of the sequence.

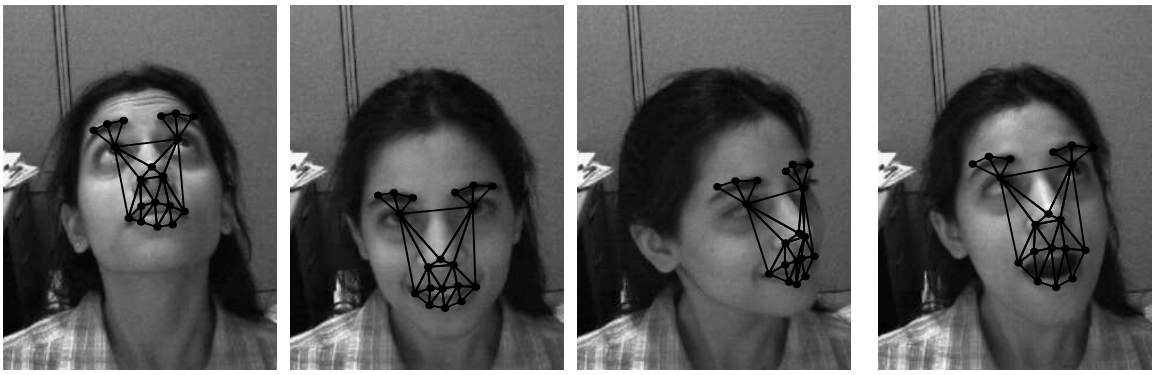


Figure 7: Monocular tracking results on subject 1

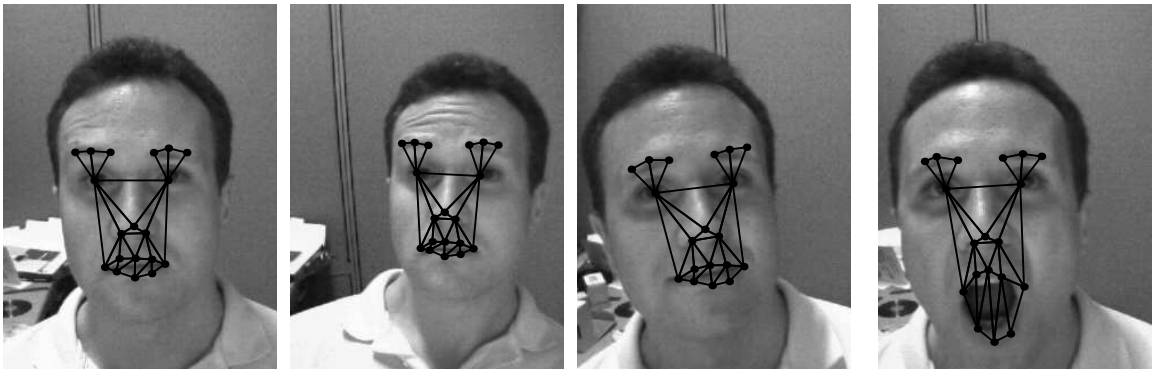


Figure 8: Monocular tracking results on subject 2

3.6 Use of Model for Face Recognition

Our face model extracts functional subspaces like shape (identity) and action (expression) as discussed in sections 3.2 and 3.4. If we estimate the shape parameter vectors for each frame in our database, we observe that the shape parameter vectors corresponding to a subject cluster together in shape space. The scatter plot in 2-dimensional σ-space is shown in Figure 5. This shows that the shape parameter vector represents the differences in face shape between people in some form. Therefore, the computed shape parameters can be useful for recognizing faces using a classifier, like SVM.

4 Remaining Work

In this section, we present the remaining work to be completed for this thesis.

4.1 Face Model

We are currently working on adding 15 nodes to the original face mesh. Most of these points are on the face edge. This new model will help us characterize jaw movement for facial expressions and speech and will also give us more unique face geometry information.

This face model will require some changes in the tracking algorithms as well which will be discussed in section 4.2.

During monocular tracking (section 4.4), sometimes the face mesh takes on unrealistic shapes. These are due to values of the action parameter vector $\alpha(n)$ that are outside a reasonable range. Therefore, we plan to statistically model the probability

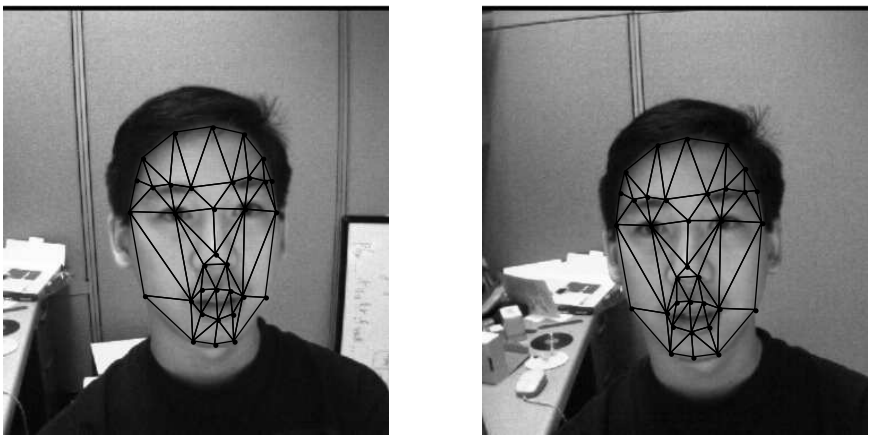


Figure 9: New Face Mesh with 34 nodes

distribution of the parameter vectors α and σ. This distribution will then be included in the model fitting algorithms, for both tracking and recognition, to constrain the values that the shape and action parameters can take.

4.2 Stereo Tracking

The algorithm described in section 3.3 sometimes has problems tracking the face boundary in the new model because the nodes on the boundary can move along the face edge. Figure 10 shows the 6th frame from the same sequence which was shown in Figure 9. The points on the face edges have travelled along the edge during the course of the five frames of tracking.

One reason for this failure is that the stereo tracking cost function (equation 4) uses image intensity and optical flow information only in a small region of interest around the nodes of the face mesh. A possible solution to this problem is to replace the region of interest intensity difference by the intensity difference between the overall face texture.

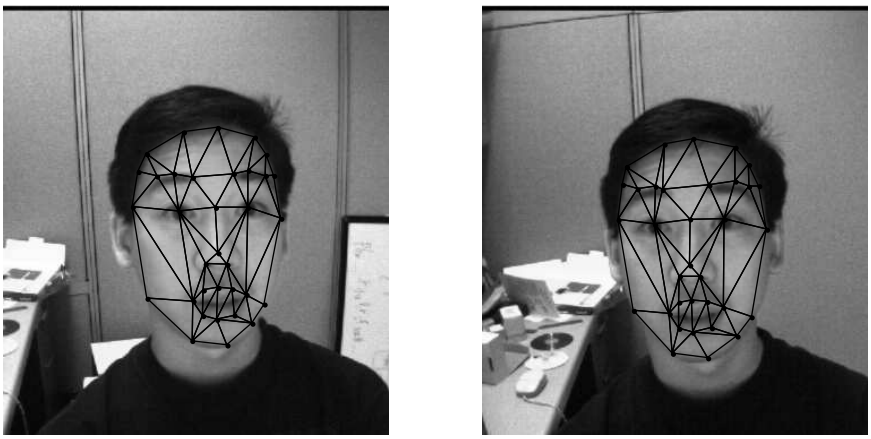


Figure 10: Failure of stereo tracking

4.3 Model Learning

Costen et al [21] use a model learning algorithm with functional subspaces in which the projection error for the shape and action subspaces is weighted by the eigenvalues to recompute the face ensembles, as discussed in section 2.2. We are working on comparing that algorithm with our model learning. In our current algorithm (see section 3.4), the neutral face shapes are recomputed every iteration by subtracting the projection of the action subspace from the data. It might be useful to weight the projection based on the eigenvalues associated with the action basis vectors.

4.4 Monocular Tracking

When using the face model to track a monocular sequence, the largest errors are usually in the rotation vector $\omega(n)$. Specifically, the error is in out-of-plane rotation angles. An example is shown in Figure 11. Even though the subject has the same head orientation in both the images, the angle of rotation estimated by the tracking is not the same as shown in the face models plotted in the figure. Figure 11(c) shows the angle of rotation estimate about the x-axis. The x-axis is the axis parallel to the

horizontal in the image, y-axis is the vertical axis and z-axis points from the camera towards the subject. It can be seen that the rotation estimate about the x-axis has a large variation even when the subject does not seem to be moving his head in the sequence. The other two rotation estimates, especially about the z-axis, have much smaller errors.

The addition of the mesh nodes on the face boundary will probably help this problem a little as it will add more rigid points to the model.

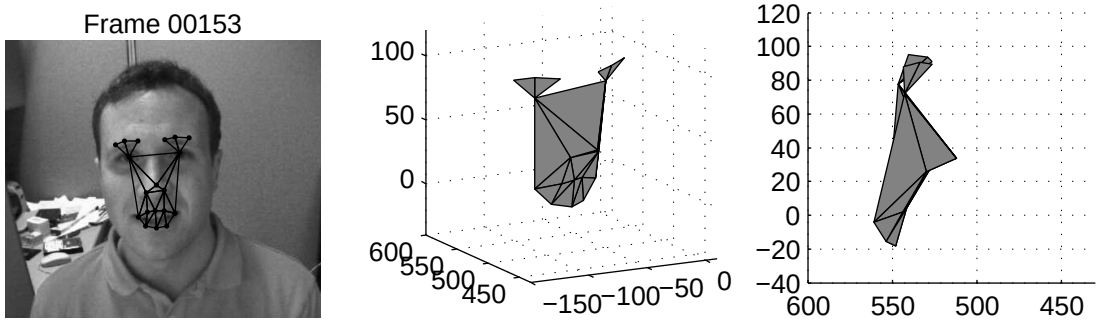
We also plan to add constraints on the tracking cost function (equation 6) based on the probability distribution of the action parameter vector. The cost function will penalize action parameter values that have a low probability, thus ensuring that such values do not occur.

4.5 Face and Expression Recognition

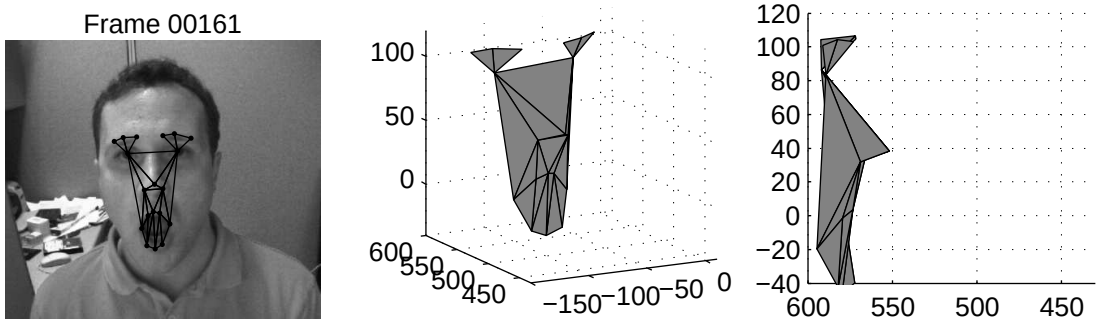
To show that the shape and action parameter vectors model only identity and facial expression respectively, we will do some experiments to recognize faces and expressions using these parameters. The purpose of these experiments is not to developed a complete and optimum recognition system but to determine how well separated the two functional subspaces (shape and action) are.

For the face recognition system, we would use Carnegie Mellon University (CMU) Pose, Illumination and Expression (PIE) database (discussed in the appendix A) because it has a larger number of subjects.

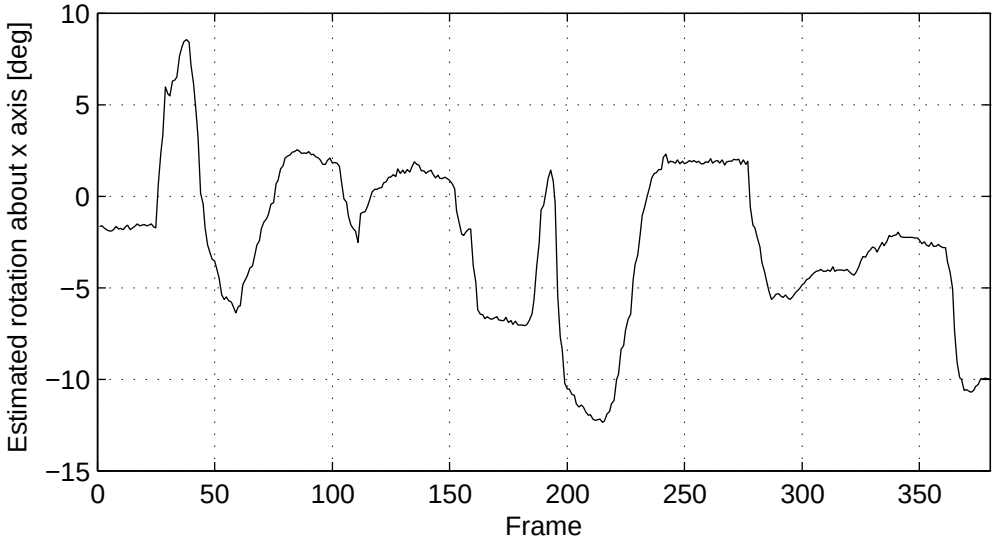
Gokturk et al [34] have done some work for facial expression recognition using an earlier version of our face model [20]. We plan to extend some of that work.



(a) Frame 153



(b) Frame 161



(c) Estimated angle of rotation about the axis connecting the eyes

Figure 11: Out-of-plane rotation errors in tracking

5 Summary

This thesis will contribute techniques for the computation and application of low-complexity face geometry models with functional subspaces.

We are proposing:

1. A better method to automatically learn a simple 3-D face model from training stereo image sequences.

2. A novel algorithm to compute functional subspaces of the face, expression and identity, simultaneously from training data [1].

3. A contribution to tracking the 3-D pose, position and facial expression in a monocular image sequence using our 3-D model.

4. Using the low-dimensional identity subspace of our model to recognize faces.

5. Recognizing facial expressions using the action subspace of our model.

A major portion of the work for items 1, 2 and 3 has been done. We are working on extending the number of nodes in the model which will require some more work in that area.

Identity and expression recognition are planned as future work.

A Experimental Setup

We are using two image sequence databases for our work: Intel face modelling database and CMU PIE database [38].

The Intel database was recorded using the Digiclops^{TM} camera system [39] at an average frame rate of 10fps, with color images of size 640×480. We acquired stereo image sequences for 11 subjects (8 male, 3 female). Four sequences per subject were recorded: two for the training set, without any rigid motion of the head; and two for testing where the subject could move his head freely. Each of the sequences was 380 frames long.

The CMU PIE (Pose, Illumination and Expression) database [38] was acquired from Carnegie Mellon University. It consists of 41,368 images of 68 people using 13 cameras. They recorded 13 poses, 43 different illumination conditions and 4 different expressions (neutral, smile, blink and talk) of each person. For our work, the most useful part of this database is the expression subset. The talking sequences are 2 seconds of video at 30fps.

References

[1] Z. Ajmal, J.-Y. Bouguet, and R. M. Mersereau, "Learning a face model for tracking and recognition," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, May 2002, vol. 4, pp. 3612–3615.

[2] R. Okada, Y. Shirai, and J. Miura, "Tracking a person with 3-D motion by integrating optical flow and depth," in *Proc. 4th IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, March 2000, pp. 336–341.

[3] S.-C. Pei, C.-W. Ko, and M.-S. Su, "Global motion estimation in model-based image coding by tracking three-dimensional contour feature points," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, pp. 181–190, April 1998.

[4] D. DeCarlo and D. Metaxas, "Deformable model-based face shape and motion estimation," in *Proc. 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, October 1996, pp. 146–150.

[5] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for medical image analysis and computer vision," in *Proc. SPIE Medical Imaging*, February 2001.

[6] J. Ahlberg, "Using the active appearance algorithm for face and facial feature tracking," in *Proc. IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, July 2001, pp. 68–72.

[7] D. DeCarlo and D. Metaxas, "The integration of optical flow and deformable models with applications to human face shape and motion estimation," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 1996, pp. 231–238.

[8] P. Ekman and W. Friesen, *The Facial Action Coding System*, Consulting Psychologist Press, Inc., 1978.

[9] P. Eisert and B. Girod, "Model-based estimation of facial expression parameters from image sequences," in *Proc. IEEE Intl. Conference on Image Processing*, October 1997, vol. 2, pp. 418–421.

[10] M. Rydfalk, "CANDIDE: A parametrized face," Tech. Rep. Report No. LiTH-ISY-I-0866, Dept. of Electrical Eng., Linköping University, Sweden, 1987.

[11] P. Eisert and B. Girod, "Analyzing facial expressions for virtual conferencing," *IEEE Computer Graphics and Applications*, vol. 18, pp. 70–78, September 1998.

[12] MPEG-4, *SNHC Verification Model 4.0, Document N1666*, April 1997.

[13] P. Eisert, *Very Low Bit-Rate Video Coding Using 3-D Models*, Ph.D. thesis, University of Erlangen-Nuremberg, Germany, November 2000.

[14] J. Ahlberg, "CANDIDE-3 – an updated parametrized face," Tech. Rep. LiTH-ISY-R-2326, Dept. of EE, Linköping University, Sweden, January 2001.

[15] J. Ahlberg, *Model-based Coding – Extraction, Coding and Evaluation of Face Model Parameters*, Ph.D. thesis, Linköping University, Sweden, September 2002.

[16] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3-D shape from image streams," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 2000, pp. 690–696.

[17] V. Blanz and T. Vetter, "Morphable model for the synthesis of 3-D faces," in *Proc. SIGGRAPH'99*, Aug 1999, pp. 187–194.

[18] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision," Draft report, Imaging Science and Biomedical Engineering, University of Manchester, UK, October 2001, `http://www.isbe.man.ac.uk/~bim/`.

[19] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. European Conference on Computer Vision (ECCV)*, 1998, vol. 2, pp. 484–498.

[20] S. B. Gokturk, J.-Y. Bouguet, and R. Grzeszczuk, "A data-driven model for monocular face tracking," in *Proc. IEEE Intl. Conf. on Computer Vision (ICCV)*, July 2001, vol. 2, pp. 701–708.

[21] N. P. Costen, T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Simultaneous extraction of functional face subspaces," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 1999, pp. 492–497.

[22] N. P. Costen, T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Automatic extraction of the face identity-subspace," *Image and Vision Computing*, vol. 20, no. 5–6, pp. 319–329, March 2002.

[23] N. P. Costen, T. F. Cootes, and C. J. Taylor, "Compensating for ensemble-specific effects when building facial models," *Image and Vision Computing*, vol. 20, no. 9–10, pp. 673–682, August 2002.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Intl. Joint Conf. on Artificial Intelligence*, 1981, pp. 674–679.

[25] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 99–127, July 2000.

[26] D. DeCarlo and D. Metaxas, "Adjusting shape parameters using model-based optical flow residuals," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 6, pp. 814–823, June 2002.

[27] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 1994, pp. 84–91.

[28] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face recognition using active appearance models," in *Proc. European Conference on Computer Vision (ECCV)*, 1998, vol. 2, pp. 581–695.

[29] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3-D morphable model," in *Proc. 5th Intl. Conference on Automatic Face and Gesture Recognition*, May 2002, pp. 202–207.

[30] S. Romdhani, V. Blanz, and T. Vetter, "Face identification by fitting a 3-D morphable model using linear shape and texture error functions," in *Proc. European Conf. on Computer Vision (ECCV)*, May 2002, pp. 3–19.

[31] J. Huang, V. Blanz, and B. Heisele, "Face recognition using component-based SVM classification and morphable models," in *Proc. 1st Intl. Workshop on Pattern Recognition with Support Vector Machines*, August 2002, pp. 334–341.

[32] B. Fasel and J. Luettin, "Facial expression analysis and recognition: A survey," IDIAP-RR 19, IDIAP, 1999.

[33] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 743–756, July 1997.

[34] S. B. Gokturk, J.-Y. Bouguet, C. Tomasi, and B. Girod, "Model-based face tracking for view-independent facial expression recognition," in *Proc. IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, May 2002, pp. 287–293.

[35] O. Faugeras, *Three dimensional Computer Vision*, MIT Press, 1993.

[36] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[37] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistal Society B*, pp. 285–339, 1991.

[38] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. 5th Intl. Conference on Automatic Face and Gesture Recognition*, May 2002, pp. 46–51.

[39] Digiclops. Point Grey Research, `http://www.ptgrey.com/products/digiclops`.