A Data-Driven Model for Monocular Face Tracking

Salih Burak Gokturk
Robotics Laboratory
Stanford, CA 94305, USA

Jean-Yves Bouguet and Radek Grzeszczuk
Intel Corporation - SC12-303
Santa Clara, CA 95052, USA

Abstract

This paper describes a two-stage system for 3D tracking of pose and deformation of the human face in monocular image sequences without the use of special markers. The first stage of the system learns the space of all possible facial deformations by applying Principal Component Analysis on real stereo tracking data. The resulting model approximates any generic shape as a linear combination of shape basis vectors. The second stage of the system uses this low-complexity deformable model for simultaneous tracking of pose and deformation of the face from a single image sequence. This stage is known as model-based monocular tracking. There are three main contributions of this paper. First we demonstrate that a data-driven approach for model construction is suitable for tracking non rigid objects and offers an elegant and practical alternative to the task of manual construction of models using 3D scanners or CAD modelers. Second, we show that such a method exhibits good tracking accuracy (errors less than 5 mm) and robustness characteristics. Third, we demonstrate that our system exhibits very promising generalization properties in enabling tracking of multiple persons with the same 3D model.

1 Introduction

Face tracking is one of the main components needed for applications in human-computer interaction, model-based compression, and video conferencing. The problem of face tracking can be divided into two parts: pose determination and facial expression recognition. Many researchers in the computer vision community have investigated these two problems in the last ten years [5, 16, 15]. This paper describes a system that tracks both the 3D pose and the shape of the human face in front of a single video camera without using any markers. The system incorporates a linearized 3D face shape model with optical flow information to obtain robust monocular tracking results.

The system is divided into two main stages as illustrated in Figure 1. In the first stage, the three dimensional deformable model of the subject's face is built using a stereo camera. For this purpose, a low complexity face mesh is initialized and then tracked using optical flow techniques. In order to handle non-rigid deformations, each point is tracked independently. Once the 3D structure of the face is determined at each frame of the stereo sequence, a compact deformable shape model is computed using Principal Component Analysis (PCA).

In the second stage, the 3D model extracted in the first stage is used for monocular tracking. In this stage, the user is free to change his/her pose and facial expression. The system uses optical flow information in a model-based manner to track the changes in the user's pose and facial expression.

By using real data to construct the deformable model, we are able to take advantage of some desirable properties. First, we only need to find a few shape basis vectors to span a variety of facial expressions such as smiling, talking, and raising of the user's eyebrows. Second, after building up a model using data collected from one or several users in our sample database, it is possible to track a new face from a user who is not in this database.

The rest of the paper is organized as follows. We first give an overview of previous work on face tracking in Section 1.1. We describe the details of our system in Section 2. In Section 3, four experiments that demonstrate the effectiveness of our model are presented. We discuss possible extensions for future work and conclude in Section 4.

1.1 Previous Work

The system uses a model-based optical flow tracking algorithm to simultaneously estimate both pose and shape parameters. A large number of researchers [12, 11, 14, 17, 2] have studied optical flow tracking. More recently, DeCarlo and Metaxas [6, 5] presented a model-based tracking algorithm in which face shape model and motion estimation were integrated using optical flow and edge information. The deformable model they use for tracking consists of several articulated parts and has to be designed by hand prior to tracking. Eisert *et al.* in [9, 10] tracked the human face in a video sequence using a similar model-based approach.

Principal Component Analysis (PCA) is a powerful method used for optimally estimating a low-dimensional representation of data embedded in a high-dimensional space. Both Bregler *et al.* [4] and Blanz and Vetter [3] have used PCA to approximate non-rigid shapes as linear combinations of sets of rigid basis shapes. However, neither of these works combines optical flow tracking with shape parameter estimation. In the context of 2D image processing, Turk *et al.* [19] demonstrated that Eigen-images can be used to characterize the appearance of faces in images. In this work, PCA was shown to be a good, practical solution to the problem of face recognition. Since then, a large number of authors have used the same approach for face recognition and detection, such as Lee *et al.* [13].

Singh *et al.* in [18] has studied the class of deformations of non-rigid objects. In this work, shape estimation was formulated as an energy minimization problem. Akgul and Kambhamettu [1] studied the general problem of tracking continuous and deformable surfaces from stereo data. Their approach used a dual communicating mesh that deforms according to internal smoothness energy and external energy coming from image forces.

0-7695-1143-0/01 $10.00 © 2001 IEEE

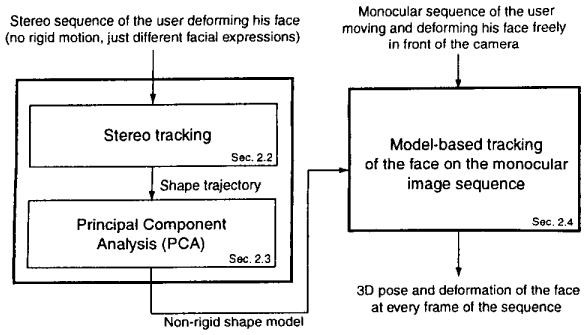

Figure 1: The main components of the algorithm.

2 Description of the System

2.1 Deformable Face Model

Here we consider the problem of tracking the human face in 3D space from a monocular sequence of images. Let I_n be the n^{th} image of that sequence. We represent the 3D structure of the face at time n by a collection of N points P_n^i $(i = 1, \ldots, N)$. We define the face reference frame as a reference frame attached to the head of the user. Let $\mathbf{X}^i(n)$ and $\mathbf{X}_c^i(n) = [X_c^i(n) \;\; Y_c^i(n) \;\; Z_c^i(n)]^T$ be the coordinate vectors of the point P_n^i in the face reference frame and camera reference frame respectively. The vectors $\mathbf{X}^i(n)$ and $\mathbf{X}_c^i(n)$ are then related to each other through a rigid body transformation characterizing the pose of the user's face with respect to the camera at time n,

$$\mathbf{X}_c^i(n) = \mathbf{R}_n \mathbf{X}^i(n) + \mathbf{t}_n, \tag{1}$$

where $\mathbf{R}_n$ is a 3×3 rotation matrix, and $\mathbf{t}_n$ is a translation vector. The problem of tracking the face as a non-rigid object corresponds to estimating simultaneously the quantities $\mathbf{X}^i(n)$ for shape, and $\mathbf{R}_n$ and $\mathbf{t}_n$ for pose. Of course, since $\mathbf{R}_n$ is a rotation matrix, it is uniquely parameterized by a 3-vector $\overline{\omega}_n$ also known as rotation vector. Rotation matrix and rotation vector are related to each other through the Rodrigues formula (see [11]).

The only data available for estimating shape and pose are the images $I_n, n = 1, 2, \ldots, M$. Let p_n^i be the projection of P_n^i on image I_n, and let $\mathbf{x}_n^i$ be the image coordinate vector of p_n^i. Following the traditional pinhole camera model, the image coordinate vector $\mathbf{x}_n^i$ for the projection of P_n^i is

$$\mathbf{x}_n^i = \begin{bmatrix} x_n^i \\ y_n^i \end{bmatrix} = \begin{bmatrix} X_c^i(n)/Z_c^i(n) \\ Y_c^i(n)/Z_c^i(n) \end{bmatrix} \doteq \pi\left(\mathbf{X}^i(n), \overline{\omega}_n, \mathbf{t}_n\right). \tag{2}$$

The tracking problem is then equivalent to inverting the projection map π for recovering 3D shape $\mathbf{X}^i(n)$ and pose $\{\overline{\omega}_n, \mathbf{t}_n\}$. Even if the image projection points $\mathbf{x}_n^i$ are perfectly localized and tracked on the images, there still exist an infinite number of shape and pose solutions that satisfy the projection equation. To lift this ambiguity, we could add an extreme and simple constraint of a totally rigid shape, in which case pose estimation becomes a trivial task given some initialization. However, since we desire to track facial deformations, we cannot make that assumption. An intermediate solution is to assume the non-rigid shape to be a linear combination of rigid shapes. Following this assumption, at any time n in the sequence, the shape coordinate vector $\mathbf{X}^i(n)$ may be written as the sum of a mean shape $\mathbf{X}_0^i$ vector and a linear combination of a small number of *known* shape vectors $\mathbf{X}_k^i$ $(k = 1, \ldots, p)$, which are the principal shape basis vectors

$$\mathbf{X}^i(n) = \mathbf{X}_0^i + \sum_{k=1}^{p} \alpha_n^k \mathbf{X}_k^i, \tag{3}$$

with $p \ll 3N$. The p coefficients α_n^k are then the only entities that allow for non-rigidity of the 3D shape. Note that if $p = 0$, then the face shape $\mathbf{X}^i(n)$ reduces to the rigid shape $\mathbf{X}_0^i$. In this sense, p is referred to as the dimensionality of the deformation space. Let $\overline{\alpha}_n$ be the vector of deformation coefficients: $\overline{\alpha}_n = [\alpha_n^1 \;\; \alpha_n^2 \;\; \ldots \;\; \alpha_n^p]^T$. Then the image projection map reduces to a function of the pose parameters $\overline{\omega}_n$ and $\mathbf{t}_n$ and the deformation vector $\overline{\alpha}_n$

$$\mathbf{x}_n^i \doteq \pi_i\left(\overline{\alpha}_n, \overline{\omega}_n, \mathbf{t}_n\right). \tag{4}$$

The monocular tracking algorithm combines then the Lucas-Kanade optical flow constraint with the specific mathematical form of the non-rigid model, given by Equation (3), for simultaneous estimation of the deformation vector $\overline{\alpha}_n$ and the pose parameters $\overline{\omega}_n$ and $\mathbf{t}_n$ at every frame. This procedure known as model-based tracking is described in Section 2.4.

In order to follow this formalism, it is necessary to compute the shape basis vectors $\mathbf{X}_k^i$. This is the preliminary stage of the system. Unlike traditional approaches requiring a manual construction of the non-rigid model [6, 10], we propose a data-driven approach where the principal shape basis vectors are derived from real 3D tracked data. For that purpose, a calibrated stereo camera is used to track in 3D the face of the user in a short sequence of approximately 100 to 150 frames. The basis vectors $\mathbf{X}_k^i$ are then computed from the tracked sequence using Principal Component Analysis (PCA) as described in Section 2.3. Details of stereo tracking are presented in Section 2.2. Observe that stereo tracking serves here as a means to capture 3D trajectory data for the purpose of shape deformation analysis. It may be thought of as the analogous of a motion capture system providing the necessary data for 3D human motion analysis. The reader only interested in shape deformation analysis and model-based monocular tracking may skip the stereo tracking section and go straight to Section 2.3.

2.2 Stereo Tracking

We propose to estimate the shape basis vectors $\mathbf{X}_k^i$, $k = 0, \ldots, p$, from real data computed from stereo tracking. A set of $N = 19$ points P^i located on the eyes (2), nose (3), mouth (8), eyebrow (6) are initialized on the first pair of images, and then tracked on both image streams. Since the objective of this stage is capturing facial deformations independently from pose, the user is asked to maintain head pose as fixed as possible throughout the sequence while making a variety of different facial expressions: open/close mouth, smile, raise eyebrow... Figure 2 shows an example of stereo

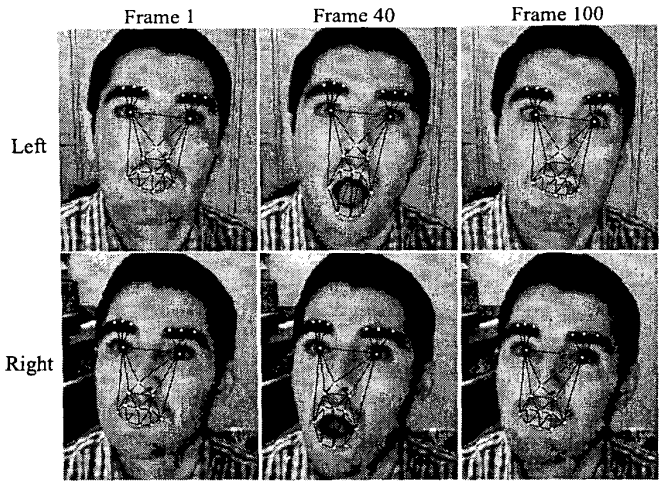

Figure 2: Stereo sequence used for shape learning.

tracking. The $N = 19$ points are manually positioned on the first left and right images and then tracked on the subsequent image pairs. Tracking is done in 3D, meaning that each point location $\mathbf{X}_c^i(n)$ (in the left camera reference frame) is updated so that its current left and right image projections best match in appearance the previous image's projections (temporal tracking). In addition, to maintain stereo correspondence throughout tracking, the left and the right image projections are constrained to match by considering an additional image cost measured between left and right images. More specifically, tracking of the point P^i from frame $n-1$ to frame n is established by minimizing a cost function $\mathbf{E}_i$ of the form

$$\mathbf{E}_i(n) = \sum_{ROI} \left\{ \begin{array}{l} \gamma_1 \left(I_n^L\left(\mathbf{x}_L^i(n)\right) - I_{n-1}^L(\mathbf{x}_L^i(n-1))\right)^2 + \\ \gamma_2 \left(I_n^R(\mathbf{x}_R^i(n)) - I_{n-1}^R(\mathbf{x}_R^i(n-1))\right)^2 + \\ \gamma_3 \left(I_n^R(\mathbf{x}_L^i(n)) - I_{n-1}^L(\mathbf{x}_R^i(n-1))\right)^2 \end{array} \right\}, \quad (5)$$

where I_n^L and I_n^R are the left and the right images at time n and the vectors $\mathbf{x}_L^i(n)$ and $\mathbf{x}_R^i(n)$ are the coordinate vectors of the left and right image projections of P^i. The summation is done over a window around the image point called Region Of Interest (ROI). The first two terms of Equation (5) are the traditional image matching costs accounting for independent left and right temporal tracking, whereas the third term maintains correspondence between right and left images. The three coefficients γ_1, γ_2 and γ_3 are fixed weighting coefficients (same for all the points) accounting for variable reliability between the three energy terms. Information on how to choose those coefficients is given in the experimental Section 3. When applied to all the mesh points, this corresponds to minimizing a global energy function $\mathbf{E}_I(n) = \sum_{i=1}^N \mathbf{E}_i(n)$. In this present form, stereo tracking works well over short sequences (20 to 30 frames), but suffers from large drifts in three dimensional space as the sequence length increases. Indeed, points being tracked totally independently are therefore free to move in any possible way with respect to each other in 3D space. Although that is desirable for capturing non-rigid deformations, it can also cause the overall 3D structure to drift to unrealistic configurations due to noise in the measurements. To remedy this problem, we propose to add to the cost function $\mathbf{E}_I(n)$ three regularization terms that force the overall 3D structure to preserve its integrity while deforming smoothly as a whole throughout the sequence. The total energy function $\mathbf{E}(n)$ to minimize becomes then

$$\mathbf{E}(n) = \mathbf{E}_I(n) + \mathbf{E}_T(n) + \mathbf{E}_S(n) + \mathbf{E}_A(n). \quad (6)$$

The temporal smoothing term $\mathbf{E}_T(n)$ minimizes the amplitude of 3D velocity at each point while the shape smoothing term $\mathbf{E}_S(n)$ minimizes differences of neighboring points' velocities. This term guarantees the integrity of the model by weakly enforcing neighbor points to move together. The anthropometric energy cost $\mathbf{E}_A(n)$ attempts to keep the segment lengths as close as possible to their values computed in the first frame, preventing drifts over long tracking sequences. A similar soft constraint was used by DeCarlo and Metaxas in [6, 5, 7]. These three regularization terms are formulated as follows:

$$\mathbf{E}_T(n) = \sum_{i=1}^N \rho_i \left\| d\mathbf{X}^i(n) \right\|^2$$

$$\mathbf{E}_S(n) = \sum_{i,j} \beta_{ij} \left\| d\mathbf{X}^i(n) - d\mathbf{X}^j(n) \right\|^2$$

$$\mathbf{E}_A(n) = \sum_{i,j} \delta_{ij} \left(\begin{array}{c} \left\| \mathbf{X}_c^i(n) - \mathbf{X}_c^j(n) \right\|^2 \\ - \left\| \mathbf{X}_c^i(1) - \mathbf{X}_c^j(1) \right\|^2 \end{array} \right)^2,$$

where $d\mathbf{X}^i(n) = \mathbf{X}_c^i(n) - \mathbf{X}_c^i(n-1)$ and the positive coefficients ρ_i, β_{ij} and δ_{ij} vary from point to point and from edge to edge. All segments $[P^i, P^j]$ that are subject to large stretches will be assigned lower β_{ij} and δ_{ij} values. Similarly, a point P^i on an highly deformable region of the face will be assigned a small ρ_i. On the other hand, points and segments that are known to be quite rigid will be assigned higher values for ρ_i, β_{ij} and δ_{ij} penalizing a lot any movement and stretch applied on them. For example, points and edges on the outline of the mouth will have lower coefficients than points and edges belonging to the nose and eyes. Numerical values for these coefficients are given in Section 3.

The solution shape $\mathbf{X}_c^i(n)$, $i = 1, \ldots, N$ that minimizes the total energy function $\mathbf{E}(n)$ may be found through gradient descent. This is done by setting the derivative of $\mathbf{E}(n)$ with respect to all the differential shape coordinate vectors $d\mathbf{X}^i(n)$ to zero $\partial \mathbf{E}(n)/\partial d\mathbf{X}^i(n) = 0$. After derivation of the Jacobian matrix, the solution for shape tracking reduces to a linear equation: $d\mathbf{X} = \mathcal{D}^{-1}\,\mathbf{e}$, where $d\mathbf{X}$ is a $3N \times 1$ column vector consisting of all N vectors $d\mathbf{X}^i(n)$ and $\mathcal{D}$ and $\mathbf{e}$ are a $3N \times 3N$ matrix and a $3N \times 1$ vector respectively.

Once $\mathbf{dX}$ is computed, the shape $\mathbf{X}_c^i(n)$ is known. The same process is repeated throughout the stereo sequence to finally obtain a complete 3D shape trajectory. Tracking is initialized by performing stereo triangulation on the first stereo pair. This process requires manual localization of the $N = 19$ points on left and right images. The next section discusses the computation of the principal shape vectors $\mathbf{X}_k^i$ from that trajectory.

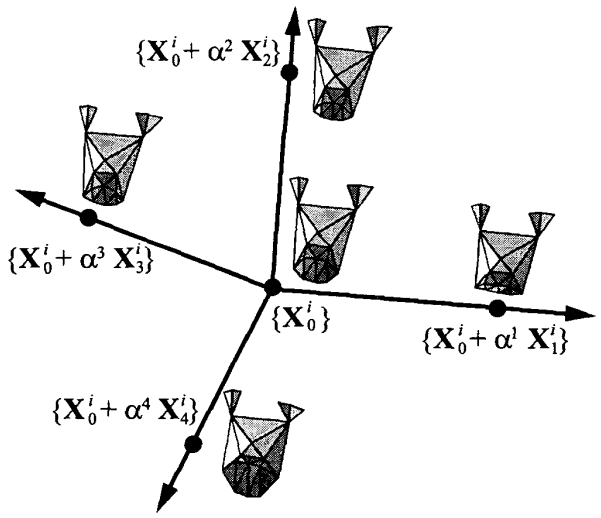

Figure 3: An illustration of the four-dimensional space of deformations learned from stereo tracking of the sequence shown in Figure 2. The mean shape $\{\mathbf{X}_0^i\}$ is shown at the center. The other four shapes are sample shapes on the four principal deformation directions $\{\mathbf{X}_0^i + \alpha^k \mathbf{X}_k^i\}$ (for this figure, $\alpha^1 = \alpha^2 = \alpha^3 = \alpha^4 = 30$cm).

2.3 Principal Shape Vectors Computation

The outcome of stereo tracking is the 3D trajectory of each point P^i in the left camera reference frame: $\mathbf{X}_c^i(n) = [X_c^i(n)\ Y_c^i(n)\ Z_c^i(n)]^T$, for $n = 1, \ldots, M$ where M is the number of frames in the sequence. The $p+1$ shape basis vectors $\mathbf{X}_k^i$ are computed using Singular Value Decomposition (SVD). First the mean shape $\mathbf{X}_0^i$ is computed:

$$\mathbf{X}_0^i = \frac{1}{M} \sum_{n=1}^{M} \mathbf{X}_c^i(n). \tag{7}$$

Next, the mean shape is subtracted from the whole trajectory: $\mathcal{X}_c^i(n) = \mathbf{X}_c^i(n) - \mathbf{X}_0^i$. The resulting shape trajectory $\mathcal{X}^i(n)$ is then stored in a $3N \times M$ matrix

$$\mathbf{M} = \begin{bmatrix} \mathcal{X}_c^1(1) & \mathcal{X}_c^1(2) & \cdots & \mathcal{X}_c^1(M) \\ \mathcal{X}_c^2(1) & \mathcal{X}_c^2(2) & \cdots & \mathcal{X}_c^2(M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{X}_c^N(1) & \mathcal{X}_c^N(2) & \cdots & \mathcal{X}_c^N(M) \end{bmatrix}. \tag{8}$$

Applying Singular Value Decomposition (SVD) on $\mathbf{M}$, we obtain $\mathbf{M} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T$ where $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_1\ \ldots\ \mathbf{u}_{3N}]$ and $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_1\ \ldots\ \mathbf{v}_M]$ are two unitary $3N \times 3N$ and $M \times M$ matrices and $\mathbf{S} = diag(\sigma_1, \ldots, \sigma_{3N})$ is the diagonal matrix of the positive and monotonically increasing singular values σ_k. Following this decomposition, we have

$$\mathbf{M} = \sum_{k=1}^{3N} \sigma_k \mathbf{u}_k \mathbf{v}_k^T. \tag{9}$$

Truncating this sum from $3N$ to p terms results in an optimal least squares approximation of the matrix $\mathbf{M}$, given a fixed budget of p vectors. This is equivalent to approximating each column vector of $\mathbf{M}$ (i.e. each 3D shape in the sequence) by its orthogonal projection onto the linear subspace spanned by the first p vectors $\mathbf{u}_1, \ldots, \mathbf{u}_p$. These vectors are precisely the remaining p deformation shape vectors $\mathbf{X}_k^i$: for $k = 1, \ldots, p$,

$$\mathbf{u}_k = \begin{bmatrix} \mathbf{X}_k^1 \\ \mathbf{X}_k^2 \\ \vdots \\ \mathbf{X}_k^N \end{bmatrix}. \tag{10}$$

This model is suitable for the monocular tracking stage, as long as the user produces the variety of facial expressions that have been exposed to the system during stereo shape learning. Observe that since the vectors $\mathbf{u}_k$ are unitary, the shape coefficients α_n^k appearing in Equations (3) and (4) are in units of the mean shape $\mathbf{X}_0^i$ which is centimeters in our case. Experimentally, we found that four vectors are enough to cover most common facial expressions (mouth and eyebrow movements). It is however subject to change from experiment to experiment especially as the diversity of facial expressions performed by the subject varies. Figure 3 gives an illustration of the four dimensional space of deformations computed from the stereo sequence shown in Figure 2. From Figure 3, we may observe that the basis shapes correspond to combinations of the four main facial movements: smile, open/close mouth, left and right raised eyebrows.

2.4 Model-Based Monocular Tracking

In its original form, optical flow tracking computes the translational displacement of every point in the image given two successive frames (see [14, 17]). Each image point is then processed independently. In the case of model-based tracking, all the points in the model are linked to each other through the parameterized 3D model (given here by Equation (3)). Following this formalism, the parameters defining the model configuration are estimated all at once from image measurements. In our case, those parameters are $\overline{\alpha}_n$ for shape and $\{\overline{\omega}_n, \mathbf{t}_n\}$ for pose. Assume that the face model has been tracked from the first frame of the sequence I_1 to the $(n-1)^{\text{th}}$ frame I_{n-1}. The objective is then to find the optimal pose $\{\overline{\omega}_n, \mathbf{t}_n\}$ and deformation $\overline{\alpha}_n$ of the face model that best fit the subsequent frame I_n. For that purpose, let us define a cost function $\mathbf{C}_n$ whose minimum is achieved at the tracking solution

$$\mathbf{C}_n = \sum_{i,ROI} \left\{ \begin{array}{c} (1-\epsilon) \left(I_n(\mathbf{x}_n^i) - I_{n-1}(\mathbf{x}_{n-1}^i)\right)^2 \\ +\epsilon \left(I_n(\mathbf{x}_n^i) - I_1(\mathbf{x}_1^i)\right)^2 \end{array} \right\} \tag{11}$$

$$\mathbf{x}_n^i = \pi_i \left(\overline{\alpha}_n, \overline{\omega}_n, \mathbf{t}_n\right), \tag{12}$$

where π_i is the model-based image projection map defined in Equation (4). The summation in Equation (11) is done over small pixel windows (ROI) around every image point $\mathbf{x}_n^i$, $\mathbf{x}_{n-1}^i$ and $\mathbf{x}_1^i$. Observe that the first term in Equation (11) is the standard matching cost used in the Shi-Tomasi-Kanade feature tracker [14, 17]: it measures overall image mismatch between two successive images at the model points. The second term however measures the image mismatch between the current image I_n and the first image

I_1. This additional term weakly enforces every facial feature to appear the same on the images from the beginning to the end of the sequence (in an image neighborhood sense). As such, it avoids tracking drifts and increases robustness. It is referred to as drift monitoring energy term. The two energy terms are weighted one relative to the other by the scalar ϵ. For all experiments, we kept $\epsilon = 0.2$ to emphasize standard tracking cost over monitoring cost. Tracking is equivalent to estimating the optimal pose and deformation update vectors $d\overline{\omega} \doteq \overline{\omega}_n - \overline{\omega}_{n-1}$, $d\mathbf{t} \doteq \mathbf{t}_n - \mathbf{t}_{n-1}$ and $d\overline{\alpha} \doteq \overline{\alpha}_n - \overline{\alpha}_{n-1}$. This is done by setting the derivative of $\mathbf{C}_n$ with respect to $d\overline{\alpha}$, $d\overline{\omega}$ and $d\mathbf{t}$ to zero

$$\frac{\partial \mathbf{C}_n}{\partial \mathbf{s}} = 0, \qquad \text{where} \qquad \mathbf{s} = \begin{bmatrix} d\overline{\alpha} \\ d\overline{\omega} \\ d\mathbf{t} \end{bmatrix}. \tag{13}$$

Equation (13) is solved for $\mathbf{s}$ while assuming small motion between two consecutive frames. Let I_{t_i} be the extended temporal derivative defined as follows

$$\begin{aligned} I_{t_i}(\mathbf{x}^i_{n-1}) =& I_n(\mathbf{x}^i_{n-1}) \\ & - \left((1-\epsilon)\, I_{n-1}(\mathbf{x}^i_{n-1}) + \epsilon\, I_1(\mathbf{x}^i_1)\right). \end{aligned} \tag{14}$$

The temporal derivative function I_{t_i} is in fact evaluated in the neighborhood of the point $\mathbf{x}^i_{n-1}$. Notice that if $\epsilon = 0$, Equation (14) reduces to the true temporal difference $I_{t_i} = I_n - I_{n-1}$. If $\epsilon > 0$, the image patch on the previous image I_{n-1} is averaged with that of the first frame (second row of Equation (14)). The resulting patch is used as reference for the next image I_n. That process effectively helps the tracking system 'remember' the original appearance of the feature as it was selected on the first image, thereby improving robustness and reducing drifts. Next, let $I_{\mathbf{x}_i}$ be x and y image derivatives (image gradient) of image I_n in the neighborhood of $\mathbf{x}^i_{n-1}$:

$$I_{\mathbf{x}_i} = \frac{\partial I_n}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial I_n}{\partial x} & \frac{\partial I_n}{\partial y} \end{bmatrix}. \tag{15}$$

Let ∇I^i be the derivative of the image brightness I_n with respect to $\mathbf{s}$ in the neighborhood of $\mathbf{x}^i_{n-1}$ at $\mathbf{s} = 0$:

$$\nabla I_i = \frac{\partial I_n}{\partial \mathbf{s}} = \frac{\partial I_n}{\partial \mathbf{x}} \frac{\partial \pi_i}{\partial \mathbf{s}} = I_{\mathbf{x}_i} \frac{\partial \pi_i}{\partial \mathbf{s}}. \tag{16}$$

Notice that the resulting matrix ∇I_i is of size $1 \times (p+6)$ since $I_{\mathbf{x}_i}$ and $\frac{\partial \pi_i}{\partial \mathbf{s}}$ are of respective sizes 1×2 and $2 \times (p+6)$. The optimal shape and pose update vector $\mathbf{s}$ that satisfies Equation (13) is then

$$\mathbf{s} = -\mathbf{G}^{-1}\, \mathbf{b}, \tag{17}$$

where the $(p+6) \times (p+6)$ matrix $\mathbf{G}$ and the $(p+6) \times 1$ vector $\mathbf{b}$ are given by

$$\mathbf{G} = \sum_{i=1}^{N} \sum_{ROI} \nabla I_i^T\, \nabla I_i$$

$$\mathbf{b} = \sum_{i=1}^{N} \sum_{ROI} I_{t_i}\, \nabla I_i^T.$$

Observe that the expression for the tracking solution $\mathbf{s}$ given by Equation (17) is similar in structure to that of the standard Shi-Tomasi-Kanade feature tracker [17]. In our case, a unique tracking solution is computed for the overall model all at once, while in its original form, each image point is processed individually. DeCarlo *et al* in [6, 5] and Eisert *et al.* in [9, 10] also reached similar expressions for their model-based tracking systems. The main difference between their systems and ours is in the particular form of 3D model used for tracking. In our case, the shape model is built from real data and parameterized with much fewer coefficients. For $\mathbf{s}$ to be computable, the matrix $\mathbf{G}$ must be of rank $p+6$. Roughly, each point in the model brings either zero, one or two scalar observation constraints depending on whether it falls in a textureless region, an edge region or a fully textured region in the images (See [17]). The total number of constraints collected over all the points must then be larger than or equal to $p+6 = 10$ to make the model "good to track". Once $\mathbf{s}$ is computed, pose and deformation are known at time frame n (in fact, for best results, we suggest to iterate the same procedure 4 or 5 times at the fixed time frame n to refine the estimate). The same overall process is then repeated over the subsequent frames. Initialization of the model parameters is done manually by first localizing by hand the $N = 19$ facial features on the first image I_1. A small optimization is then performed for computing the initial pose and deformation parameters $\{\overline{\omega}_1, \mathbf{t}_1, \overline{\alpha}_1\}$ that make the image projection of the model match the manually selected points.

Note that the region of interest of each model point is not kept constant throughout the sequence. Instead, its size and geometry are recomputed at every frame based on the distance (depth) and orientation (local surface normal) of the point in space. The resulting regions of interest are small parallelograms as shown in Figure 7. In particular, points that face away from the camera are declared "non visible", have a zero-size region of interest assigned to them, and therefore do not contribute to the tracking update. This method lets us handle model self-occlusions naturally.

3 Experiments

In this section, we present and discuss experimental results for the two main stages of the system: stereo tracking, and monocular model-based tracking. Figure 3 shows the 19 point mesh that was used in all of the experiments. Figure 2 shows a number of sample images with the face mesh superimposed on them. Observe that all the points do not need to fall in textured areas of the image. This is a requirement for independent feature point tracking (to declare a point "good to track" [17]), but not for model-based tracking. For example, the point at the tip of nose falls in a totally textureless region, and the points on the outline of the mouth and on the eyebrows are edge features. All those points would be impossible to track individually using traditional optical flow techniques. For stereo data acquisition, the Digiclops™ camera system [8] was used at an average frame rate of 4fps, with color images of size 640×480. We tested our system on four different users, acquiring two sequences for each

one: one stereo sequence for shape learning and one monocular sequence for model-based tracking. In this paper, we report results of two of the four users. Performance for the remaining two test cases are similar. Finally, we present an experiment in which the face model is built using the combined stereo tracking data from two distinct persons, and then used to monocularly track a third person's face. This experiment illustrates the generalization properties of our system.

3.1 Stereo Tracking and Shape Learning

Stereo tracking was performed on four sequences corresponding to four different subjects. For all sequences, initialization was done manually as described in Section 2.2. Due to slight change of illumination between left and right images, tracking within a sequence, i.e. successive left frames or successive right frames, is more reliable than tracking between right and left frames. Therefore, the value of γ_3 is kept smaller than γ_1 and γ_2. The ratios γ_1/γ_3 and γ_2/γ_3 are typically kept to 20. The ratio of ρ, β and δ to γ_1 and γ_2 depend mainly on the size of image patches around each model points. The values of these parameters have been hard-coded separately for each of the 19 points on the face mesh. For that purpose, each connected pair of points in the face mesh has been considered separately. Typical values of γ_1, γ_2, γ_3, ρ, β and δ are 1, 1, 0.05, 20000, 20000 and 100 for an average area of image feature patch of approximately 100 pixels (i.e. patches of size 10×10). In practice, we found that tracking is not very sensitive to the choice of the tuning parameters. We choose to judge the quality of tracking by visual inspection, deciding that 3D tracking is successful whenever all model points reproject properly on both left and right images throughout the sequence. Figure 2 shows tracking results achieved on a 150 frames long sequence. Following the tracking method described in Section 2.2, we were able to successfully process all four stereo sequences (each one consisting of an average of 150 frames). The maximum reprojection error of the model onto the left and right images is estimated to less than 2 pixels.

Figure 3 gives an illustration of the four dimensional space of deformations computed from the stereo sequence shown in Figure 2.

3.2 Monocular Tracking

The four deformation models computed for the four subjects are then used for monocular tracking. Although a single camera image stream is needed for monocular tracking, we decided to use the same stereo camera for acquisition with that used for stereo tracking. This gave us the option to run monocular tracking on the two (left and right) sequences independently, and compare the results for accuracy assessment. The accuracy figures presented in this section were derived following this approach in the form of errors in x, y and z locations of all the points. Note the x and y axes are parallel to the image plane, while z is the depth axis.

Figure 4 shows five images from our first test sequence consisting of 100 frames, i.e. 20-25 seconds of acquisition at 4fps. Throughout the sequence, the user is rotating and translating his head covering a working volume of approximately 10cm $\times$ 10cm $\times$ 10cm while opening the mouth, smiling and raising the eyebrows. We compute the root mean square (rms) error in x, y and z directions for all 19 model points at every frame. The average rms error (computed over all the frames) is estimated to 0.1 cm, 0.04 cm and 0.2 cm in x, y and z directions respectively. The maximum rms error is 0.25 cm, 0.1 cm, 0.4 cm in x, y and z directions. On average, the user was standing 60 cm away from the camera.

Figure 5 shows five images of our second test sequence consisting of 120 frames. In this sequence, the user moves with large translational motion in the x direction while rotating his head about the vertical axis (y) and changing his facial expression (open/close mouth, smile). Amplitudes of motions are approximately 25 cm. The average rms error (computed over all the frames) is estimated to 0.3 cm, 0.05 cm and 0.2 cm in x, y and z directions respectively. The maximum rms error is 0.5 cm, 0.1 cm, 0.7 cm in x, y and z directions.

For each of the two previous experiments, monocular tracking is done using the deformable model associated to the specific user to track. In other words, we assume a separate model per subject. In this format, adding a new user means having to acquire a new stereo sequence and process it for building a new deformable model. This is rather impractical. Instead, it would be preferable to construct a unique face model that would fit anyone. In other words, is there a way to build a "model that fits all?" In an attempt to answer this question we ran the following experiment. First, we built a long stereo trajectory sequence by concatenating the two tracked stereo trajectories acquired in the two first experiments. This process consists of registering the second trajectory to the first one in 3D pose space in order to best align the two eye points and the point at tip of the nose. The resulting 3D trajectory sequence contains then a rich set of facial deformations previously exposed by both users. A new four-dimensional deformable model is then derived from that data following our standard PCA approach and used to track monocularly a third person's face. Tracking results are shown in Figure 6. After close inspection of the images on this figure, one might notice that the face geometry of the third individual is slightly different from that of the other two individuals from whom the model was learned, especially in the nose area (look at the tip of the nose on frame 48). Nevertheless, 3D tracking is successful, even when the user faces away from the camera (frames 24 and 48). The average rms error for this experiment is 0.28 cm, 0.13 cm, 0.86 cm in x, y and z. The maximum rms errors are 0.43 cm, 0.29 cm and 2.41 cm in x, y and z directions respectively. Following this experiment, we believe that our system has good generalization properties exploitable for constructing a generic face model that fits all.

Figure 7 demonstrates the effectiveness of the monitoring cost term in recovering from lost tracks. On that 165-frame long sequence, tracking is lost at around frame 105, and then recovered by frame 135. This recovery is possible thanks to the additional monitoring cost of Equation (11).

The small window around each point is the region of interest around that point. Observe that the windows are not strictly rectangular and identical throughout the sequence, as explained in Section 2.4.

In spite of the low complexity model used for tracking (only $p = 4$ shape coefficients), we achieved tracking accuracies comparable to that reported by DeCarlo *et al* in [6, 5]. We credit this mainly to the fact that, at equal complexity, a model derived from a real data sequence has better chances to fit a new sequence than one designed by other means (such as CAD modelers).

4. Conclusion and Future Work

In this paper, we presented a two-stage system for 3D tracking of pose and deformation of the human face in monocular image sequences without the use of special markers. The first stage of the system learns the spaces of all possible facial deformations by applying Principal Component Analysis on real stereo tracking data. The resulting model approximates any generic shape as a linear combination of shape basis vectors. The second stage of the system uses this low-complexity deformable model for simultaneous estimation of pose and deformation of the face from a single image sequence. This stage is known as model-based monocular tracking. There are three main contributions of this paper. First we demonstrated that a data-driven approach for model construction is suitable for 3D tracking of non-rigid objects and offers an elegant and practical alternative to the task of manual construction of models using 3D scanners or CAD modelers. Second, we showed that creating a model from real data lets us represent a large variety of facial deformations with many less parameters than handcrafted models and leads to increased robustness and tracking accuracy. Third, we demonstrated that our system exhibits very promising generalization properties in enabling tracking of multiple persons with the same 3D model. This constitutes a major improvement over most other face tracking systems that require a different model for each user to track. Experimental results showed that monocular tracking is achieved to an accuracy of less than 5 mm in 3D position of all the facial features, over image sequences longer than 100 frames. In addition, we demonstrated that monocular tracking does not suffer from drifts by showing that even if the face is lost on a few frames, it is recovered in the subsequent frames.

There are many possible directions for future work. First, we intend to work on a real time implementation of the monocular tracking stage. For that purpose, we will need to develop a method for automatic initialization. The study of the correspondences between deformation parameters $\overline{\alpha}$ and facial expressions is another problem for future investigations. It is especially interesting for extending our system to 3D facial expression recognition, and lip reading. In relation to that problem, we intend to carry a formal study of the dimensionality of the facial deformation space from tracked data. This is essential for choosing the appropriate number p of principal shape basis vectors. Future work also includes investigations of methods for building the deformable model during monocular tracking. Finally, we would like to test our system for tracking other non-rigid objects.

References

[1] Y.S. Akgul and C. Kambhamettu. Recovery and tracking of continuous 3D surfaces from stereo data using a deformable dual-mesh. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:765–772, 1999.

[2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43–78, 1994.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. *Proceedings SIGGRAPH'99*, pages 187–194, 1999.

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. *Proceedings CVPR '00*, 2:690 –696, 2000.

[5] D. DeCarlo and D. Metaxas. Deformable model-based face shape and motion estimation. *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 146 –150, 1996.

[6] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. *Proceedings CVPR '96*, pages 231–238, 1996.

[7] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. *Proceedings SIGGRAPH'98*, pages 67–74, 1998.

[8] Digiclops Stereo System. Point Grey Research. Visit `http://www.ptgrey.com/products/digiclops`.

[9] P. Eisert and B. Girod. Model-based facial expression parameters from image sequences. *Proc. IEEE International Conference on Image Processing ICIP-97, Santa Barbara, CA, USA*, 2:418–421, October 1997.

[10] P. Eisert and B. Girod. Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications, Special issue: Computer Animation for Virtual Humans*, pages 70–78, September 1998.

[11] O.D. Faugeras. *Three dimensional vision, a geometric viewpoint*. MIT Press, 1993.

[12] Berthold K.P. Horn. *Robot Vision*. MIT Press, 1986.

[13] Sang-Jean Lee, Sang-Bong Jung, Jang-Woo Kwon, and Seung-Hong Hong. Face detection and recognition using PCA. *TENCON 99. Proceedings of the IEEE Region 10 Conference*, 1:84–87, 1999.

[14] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Conf. on Art. Intell.*, 1981.

[15] R. Okada, Y. Shirai, and J. Miura. Tracking a person with 3-D motion by integrating optical flow and depth. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 336–341, 2000.

[16] Soo-Chang Pei, Ching-Wen Ko, and Ming-Shing Su. Global motion estimation in model-based image coding by tracking three-dimensional contour feature points. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8:181 –190, April 1998.

[17] Jianbo Shi and Carlo Tomasi. Good features to track. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 593–600, 1994.

[18] A. Singh, D Goldgof, and D Terzopoulos. *Deformable Models in Medical Image Analysis*. IEEE Press, 1998.

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

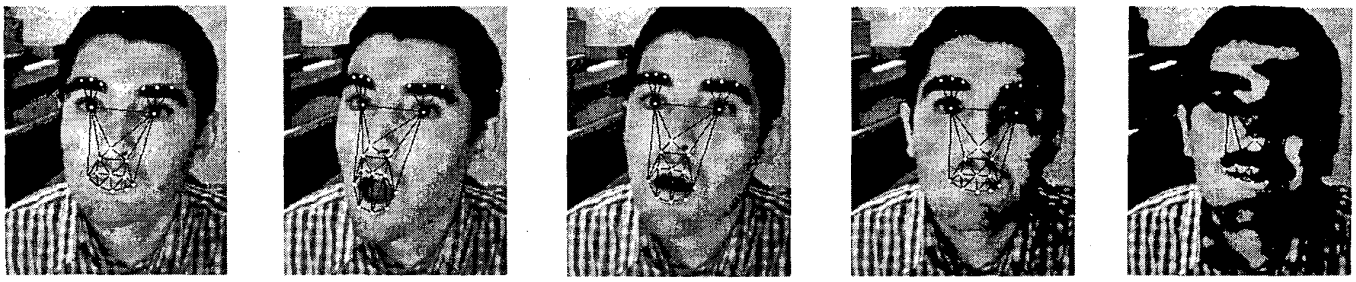

Figure 4: First monocular tracking experiment. Length of sequence: 100 frames.

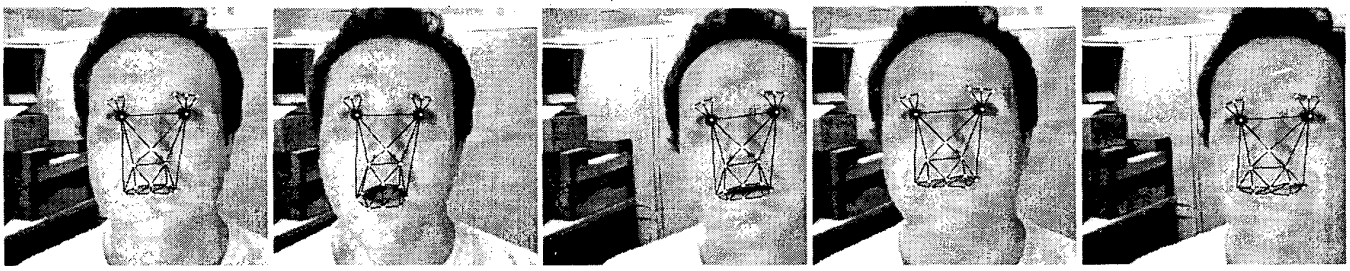

Figure 5: Second monocular tracking experiment.

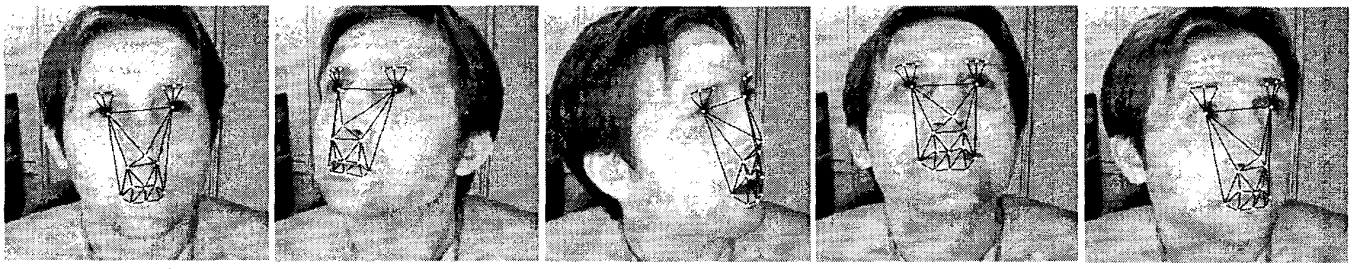

Figure 6: Third monocular tracking experiment. Tracking is done using the model built from stereo tracking data of two other persons.

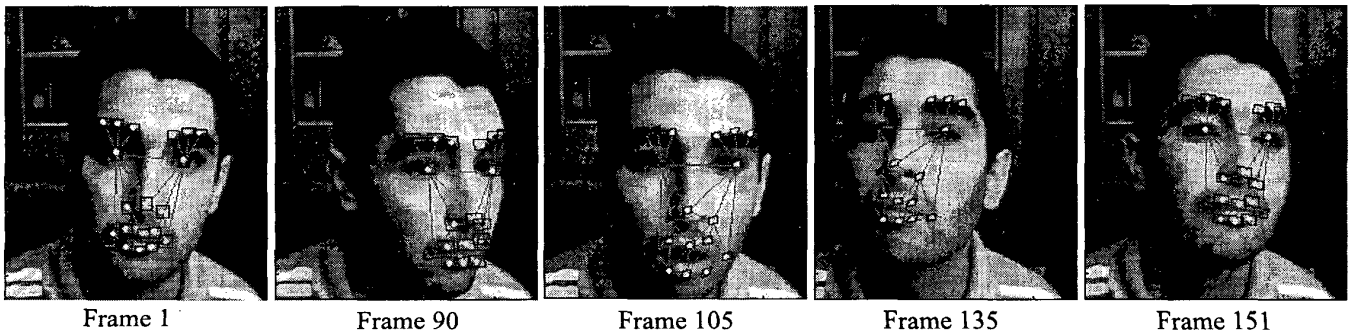

Figure 7: Fourth monocular tracking experiment. Tracking is lost at frame 105 and recovered at frame 135 thanks to the drift monitoring cost function. Length of sequence: 160 frames.